

仮想電子辞書システムにおける GUI の設計と実装

野村 智彦[†] 近藤 幹大[†] 都司 達夫[‡] 樋口 健[‡]

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

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

あらまし 我々は XML 化された電子辞書に基づく「仮想電子辞書システム」を提案している。このシステムでは、仮想辞書の定義は、見出し語からのタグの連鎖によるパス式を用いて各辞書項目を指定する必要がある。また、仮想辞書の検索文は辞書の論理構造を一次的に記述した上で検索条件と検索項目を指定する必要がある。辞書構造やその項目をこのように一次的に指定する入力操作は非常に煩わしい。そこで、ここでは、実体辞書や仮想辞書の論理構造を二次元テーブルとして Web ページ上で定義するための GUI および二次元テーブルで表現された辞書構造上で検索文を入力できる GUI および各種辞書操作を仮想辞書サーバと連携して行うための GUI を設計・実装した。

キーワード XML, 問い合わせ処理, 辞書, 情報検索, ユーザインターフェース

Implementation and Design of GUI in Virtual Electronic Dictionary System

Tomohiko NOMURA[†] Mikihiro KONDOU[†] Tatu TSUJI[‡] and Ken HIGUCHI[‡]

[†] Graduate course of Engineering, Fukui University Graduate School

[‡] Faculty of Engineering Fukui University 3-9-1 Bunkyou, Fukui-shi, Fukui, 910-0017 Japan

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

Abstract We have proposed *virtual dictionary system* based on XML formatted dictionaries. In this system, each item of a virtual dictionary needs to be defined in terms of the corresponding lengthy path expression of the real dictionary. Moreover, a retrieval query against a virtual dictionary should be specified on a linearly arranged logical structure of the dictionary. It is essential for the easy manipulation of virtual dictionaries to provide a high quality GUI. In this paper, we describe a GUI that defines the logical structures of dictionaries expressed by two dimensional graphical tables and specifies retrieval queries on these tables. The GUI also initiates other kinds of operation in combination with the virtual dictionary server.

Keyword XML, Query Processing, Dictionary, Information retrieval, User interface

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。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。

我々は XML 化された電子辞書に基づく「仮想電子辞書システム」を提案している [3]。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。辞書の構造化に関する標準化として、EPWING[2]があり、それに従った辞書も出回っているがこのような試みは見られない。

また、仮想辞書を定義する DEFINE 文(図 2)には、ルートタグから始まる絶対パス式の入力が必要であり、入力が非常に面倒である。そして、辞書の検索文において、各項目に埋め込んでいく形で検索条件と検索項目を設定している。このような仮想辞書システムを Web 上で利用するためには、辞書の論理構造を簡単に入力でき、かつ分かり易く構成要素のレベルで操作できる GUI が必須となる。

本研究では、この仮想電子辞書システムに辞書の論

2つの実体辞書上の仮想辞書

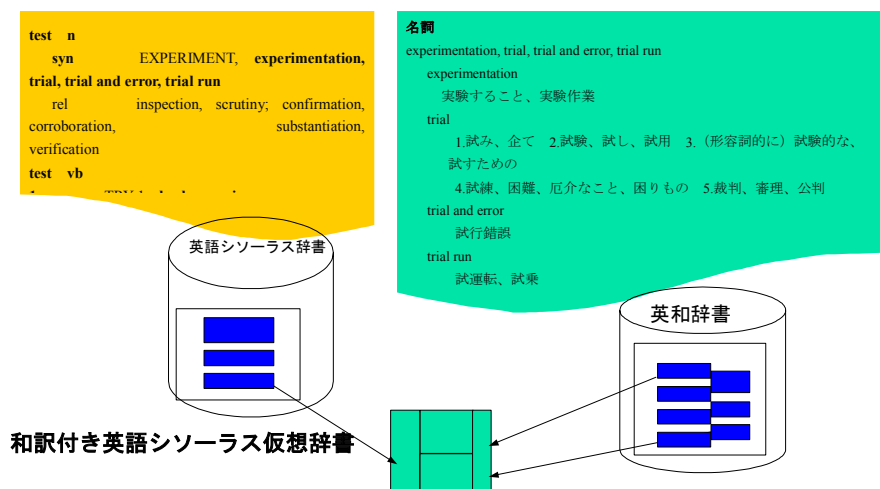


図 1 仮想辞書とは？(和訳付き英語シソーラス辞書)

理構造を構成要素のレベルで簡単に操作できる GUI を設計・実装した。

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(図 1)。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、英語シ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「和訳付き英語シソーラス辞書」を紹介する。この辞書は英語シ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)。

ここで、図 2 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。これを仮想辞書定義文(DEFINE 文)という。DEFINE 文は 5 つの句からなっている。それぞれについて説明する。

```

DEFINE
  和訳付き英語シソーラス辞書(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))))
FROM 英語シソーラス辞書, 英和辞書
ASSIGN
  見出し語 = thesaurus.midsashi,
  見出し語項目.品詞 = thesaurus.midashi_lv.part,
  見出し語項目.品詞項目.類義語 =
    thesaurus.midashi_lv.part_lv.sense_lv.syn
  見出し語項目.品詞項目.類義語項目.語義 =
    eiwa.midashi.part_lv.sense
CONNECTED BY
  eiwa.midashi = thesaurus.midashi_lv.part_lv.sense_lv.syn
WHERE eiwa.midashi_lv.part = thesaurus.midashi_lv.part
  
```

図 2 和訳付き英語シソーラス辞書の DFIEIN 文

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
となる。母体辞書（この例では英語シソーラス辞書

和訳付き英語シソーラス辞書(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))))

《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、結合条件(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。

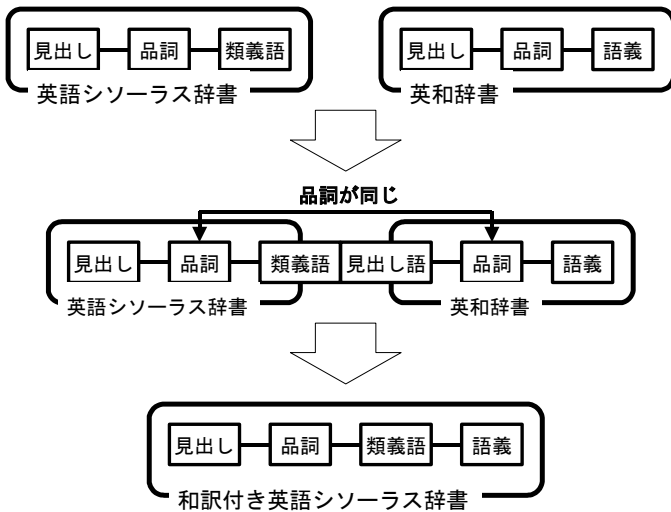


図 3 和訳付き英語シソーラス辞書の定義

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。

2.4. 辞書データの XML 化

```
<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>
```

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[1]の文書型定義(DTD)を記述して、辞書データのXML化を行った。例として、英語シソーラス辞書の見出し語「test」に対してXML化したものの一部分を以下に記す。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のことを具体項目という。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クタと呼ぶことにする。

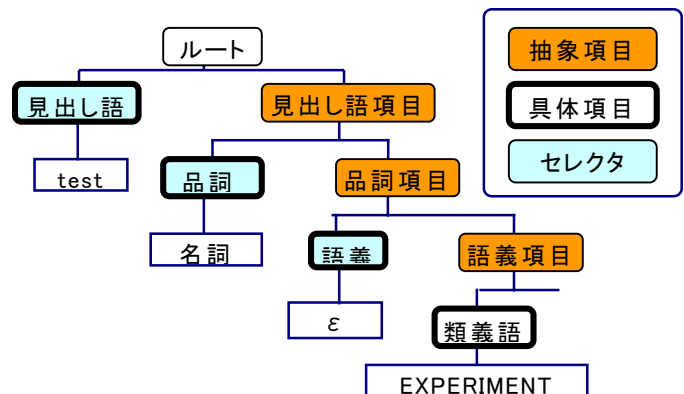


図 4 抽象項目、具体項目とセクタ

3. 仮想辞書システム

本システムは、GUI、仮想辞書サーバ、実体辞書サーバの3つから構成される(図5)。GUI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い、結果を GUI に返却する。実体辞書サーバはネットワーク上で複数存在し得る。XML による実体辞書本体を管理し、実際の検索を行う。仮想辞書サーバは実体辞書および仮想辞書の定義情報を管理しており、ネットワーク上で唯一である。

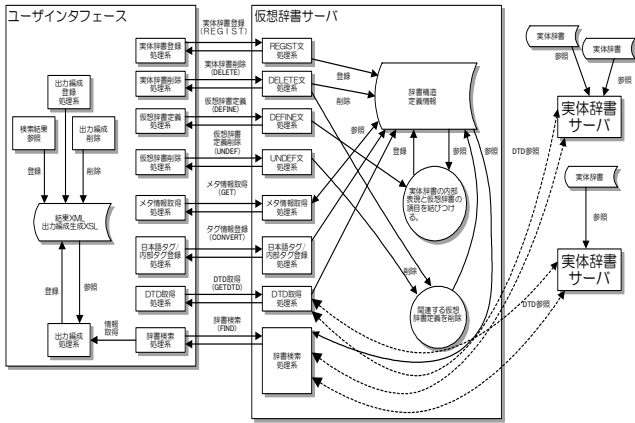


図 5 仮想電子辞書システムの構成

3.1. 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。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
- 実体辞書検索・仮想辞書検索
- メタ情報の要求処理
- 内部タグ/外部タグの登録
- 実体辞書の DTD 要求処理

3.2. 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 XML 形式の検索文を受け取る。検索文は対象辞書の 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。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。

検索文で所望のものを検索して、結果を返す。実体辞書の検索には XML パーサである 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ため、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。

る。

4. GUI

例えば、辞書構造を定義する DEFINE 文(図 2)には、ルートタグから始まる絶対パス式の入力が必要であり、入力が面倒である。仮想辞書システムを Web 上で利用するためには、辞書の論理構造を簡単に入力でき、かつ分かり易く操作できる GUI が必須となる。

そこで、辞書の論理構造表現として、HTML の TABLE タグを用いた二次元テーブルで表現している。

二次元テーブルで表現することで、辞書の論理構造が一目でわかり、各項目にコントロールを配置することにより長いパス式の入力をするこなしに定義操作や検索操作を行うことができる。

今回設計した GUI から「実体辞書の登録」「実体辞書の削除」「仮想辞書の定義」「仮想辞書の削除」「メタ情報の取得」「辞書の検索」「検索結果の出力編成」の 7 つの操作を行うことができる。

4.1. 実体辞書の登録

```
command=REGIST&option=
REGIST
英語シソーラス辞書
DICPLACE
foo, jp/VS/Rdic/thesaurus.xml
RDICSV
foo, jp/VS/Rdic/rdic.cgi
DEFINE
英語シソーラス辞書(見出し語, 見出し語項目(品詞, 品詞項目(語義,
語義項目(用例, 類義語, 関連語, 成句, 反対語, 反意語)))
GUI
英語シソーラス辞書. 見出し語:textbox
英語シソーラス辞書. 見出し語項目. 品詞:listbox(名詞, 動詞, 形容詞, 副詞, 代名詞, 接
続詞, 間投詞)
英語シソーラス辞書. 見出し語項目. 品詞項目. 語義:textbox
英語シソーラス辞書. 見出し語項目. 品詞項目. 語義項目. 用例:textbox
.
.
.
CONVERT
thesaurus=英語シソーラス辞書
midashi=見出し語
midashi_nv=見出し語項目
part=品詞
.
.
.
TABLE
part
NOUN=名詞
VERB=動詞
ADJ=形容詞
.
.
.
```

図 6 英語シソーラス辞書の REGIST 文

実体辞書の登録は図 6 のような REGIST 文によって行う。REGIST 文は、辞書名、辞書ファイルの URI、辞書が置かれている実態辞書サーバの URI、辞書の論理構造、辞書の検索時に検索条件としての具体項目値をリストボックスから選択するか、テキストボックスに直接入力するかを各具体項目に対して選択していく GUI 文、システム内で使われる内部タグと GUI で使う外部タグの相互変換テーブル(CONVERT 文)、GUI 文でリストボックスとして指定した項目のリスト値の内部

表現／外部表現変換テーブル(TABLE 文)の 8 つの情報からなっている。

実体辞書には新規に登録する DTD 付きの XML 辞書および仮想辞書の実体化によって生成される XML 辞書がある。この DTD は実体辞書の論理構造の生成に使用される。

新規登録の場合、GUI では、まず辞書名と辞書が置かれている場所、実体辞書サーバが置かれている場所を入力する。続いて、"GUI 文"を生成する。DTD の取得には GUI は仮想辞書サーバを通して、辞書の DTD を取得するコマンドを発行する。しかし、取得した DTD では、タグ名が実体辞書での内部表現となっているので、ユーザにとって分かり易い外部タグ名で取り扱えることが望ましい。そこで、内部タグ名と外部タグ名の相互変換を行うための変換テーブル(CONVERT 文)を作成する。GUI では、辞書の論理構造を二次元テーブルで表現し、このテーブル中の各項目に配置されたテキストボックスに日本語表現を入力する(図 7)。

次に GUI で、各項目に対して検索時の入力コントロールを決定する。この時、リストボックスとして指定された項目については、そのリスト値を検索する検索文を発行し、リスト値を得る。次に、リスト値の日本語変換テーブルを作成するための画面(図 8)によって TABLE 文を生成する。生成された TABLE 文と先に決定した検索時の入力コントロールの情報を使って GUI 文を生成する。これらの情報をまとめて、実体辞書登録文(REGIST 文)(図 6)として、仮想辞書サーバへ送信して、実体辞書の存在を認識させる。

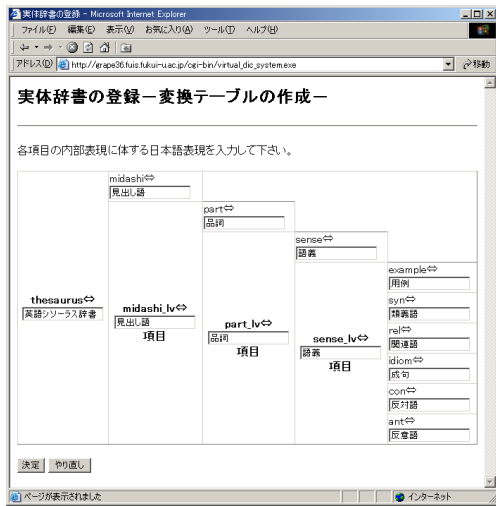


図 7 内部タグ／外部タグ変換テーブルの作成画面

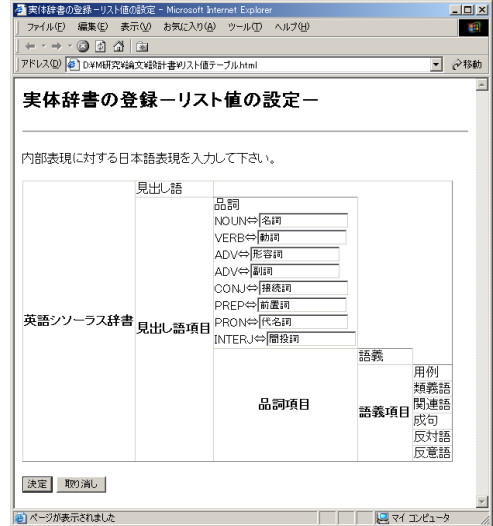


図 8 リスト値の変換テーブルの作成画面

また、仮想辞書の実体化は、仮想辞書リストから辞書名を選択し、辞書を置く場所とこの辞書を管理する実体辞書サーバの場所を指定するのみである。

4.2. 仮想辞書の定義

仮想辞書の定義は、辞書の論理構造、母体辞書、各項目と母体辞書の各項目の対応、結合項目、結合条件の 5 つの情報を決定することで定義される。新規の仮想辞書定義と既存の仮想辞書定義変更の 2 種類が考えられる。新規仮想辞書定義では、まず、辞書名の入力、母体辞書の選択を経て、論理構造定義へ移る。

GUI では、図 9 のような辞書名のみ初期状態から、仮想辞書の階層構造を順次形成していく。操作リスト(項目の変更、項目の挿入、項目の切り取り、項目のコピー、項目の入れ替え、項目名の変更、操作の終了)から操作を選択し、操作対象の項目にチェックする。また、それ以外に、操作固有の情報も入力する(例えば、項目の挿入では、抽象項目か具体項目かの指定)。

新たな項目の挿入にあたっては、母体辞書リストから母体辞書を選択(図 10)し、母体辞書の論理構造の 2 次元テーブルが表示される。仮想辞書の各項目に対して母体辞書の 2 次元テーブルの項目を GUI 上でマウントすることにより対応付けを行う(図 11、図 12)。特に母体辞書における抽象項目以下の部分構造を定義中の仮想辞書の部分構造として一度にマウントすることができる(図 13)。これを繰り返し、論理構造を完成させる。このようにして、論理構造定義と母体辞書との対応付けを同時に行うことができる。

最後に結合項目を決め、結合条件の入力を行う。一般に、結合条件は '=' による関係式の論理演算式であるが、入れ子の複雑な論理演算式を入力するのは非常に面倒である。そこで、この論理演算式を簡単に入力す

るために、論理式の文法を定義し、それに従って、まず、論理演算式を作成していく(図 14)。例えば、図 15 で論理式を選択すると図 16 のようになる。これは、図 14 の 2 番目の生成規則が適用された図である。最終的に例えば図 17 のように条件式と論理演算子のみになると、図 18 のような論理構造の 2 次元テーブルから、項目を選択し、条件式を論理演算式に埋め込んでいく。これにより、面倒なパス式の入力を誤りなく行える。

最後に、今までの情報を仮想辞書定義文として仮想辞書サーバへ送信し、仮想辞書の定義が完了する。

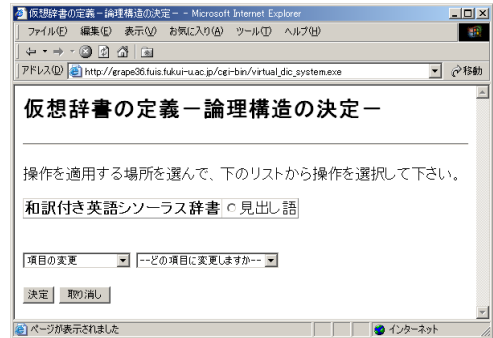


図 12 論理構造定義画面④

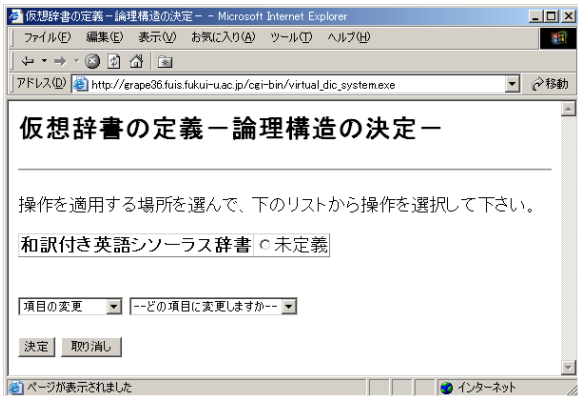


図 9 論理構造定義画面①

4.3. 辞書の検索

仮想辞書の検索指定は基本的にはセレクトア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われる。二つの検索指定法がある。スキーマレベル検索指定では、セレクトアの値に関わらず検索項目は同一である。セレクトアのすべての値について、全く同じ検索を行うことになる。これに対してインスタンスレベル検索指定とは、セレクトアの値毎に異なる検索項目が指定できる検索方法である。

まず、検索する仮想辞書を選択する。この時、選択した仮想辞書の論理構造テーブルが表示される(図 19)。検索条件と検索項目はこの中に埋め込まれているコントロール(テキストボックス, チェックボックス, リストボックス etc) を使うことで入力していく。

検索法に応じて、以後の入力方法が異なる。スキーマレベル検索を行う場合、①品詞など値が少数で列挙できるセレクトアについてはリストボックスによる選択で行う。②値が多いセレクトアの場合には、テキストボックスを使って、値そのものまたは値の正規表現が入力可能となっている。検索条件を指定した後、セレクトアに対応する抽象項目の構造の中で検索項目を指定する。インスタンスレベル検索の場合、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。この時、品詞セレクトアで名詞を選択し、類義語にチェックを入れる。次に、品詞セレクトアの選択を動詞に変更し、類義語と語義にチェックを入れる。入力確認や変更時には、リストボックスで選択する。また、検索項目を選択することを容易にするボタンが各抽象項目に配置されている。これにより、抽象項目以下の具体項目全てを検索項目に設定することが簡便に行える。図 19 は検索条件を見出し語セレクトア「test」、品詞セレクトア「名詞」で、検索項目を類義語、語義とする画面である。この FIND 文は図 20 のようになる。

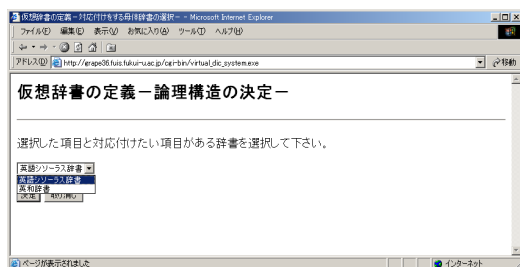


図 10 論理構造定義画面②

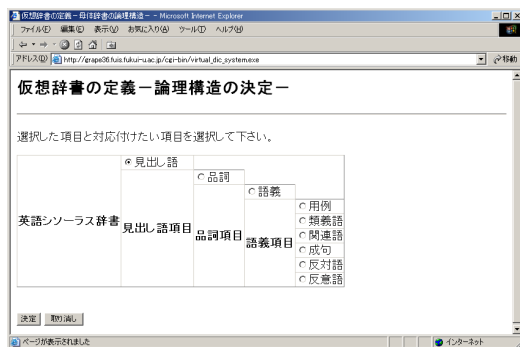


図 11 論理構造定義画面③

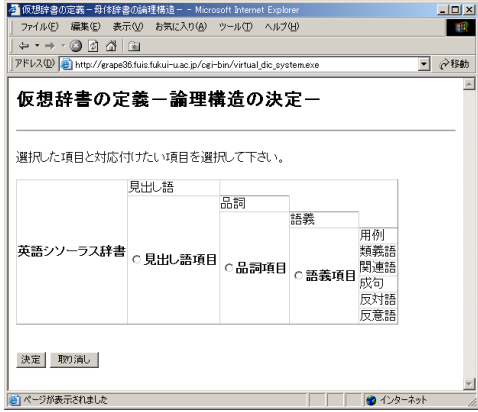


図 13 母体辞書の論理構造画面

(E は論理演算式、T は '=' による関係式)

$$G = \begin{cases} E \rightarrow T \\ E \rightarrow E \text{ LOGICAL_OPERATOR } T \\ T \rightarrow (E) \\ \text{LOGICAL_OPERATOR} \rightarrow \& \& \\ \text{LOGICAL_OPERATOR} \rightarrow | | \end{cases}$$

図 14 論理演算式の文法

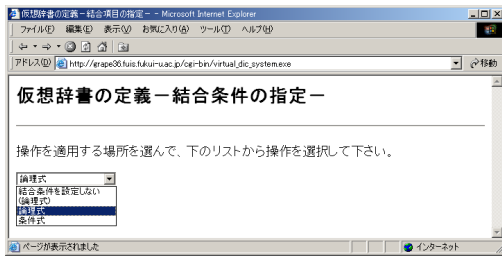


図 15 結合条件の指定画面①

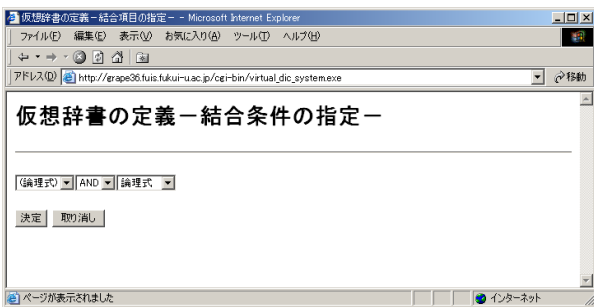


図 16 結合条件の指定画面②

4.4. 出力編成

検索結果に対してユーザの好みの出力が得られるような XSL(Extensible Stylesheet Language)を自動生成する。本 GUI では、検索結果の出力フォーマットをユーザが決めることができる。辞書の検索時に、検索条件の設定と同時に行う。

検索結果の表示方法の選択するためのリストボッ

クスを選択することで、辞書の論理構造に沿って表示するか、検索項目のみをリスト表示するかの選択が可能である。また、出力編成の設定を行わず、デフォルトの表示方法で検索結果を表示することもできる(図 22)。

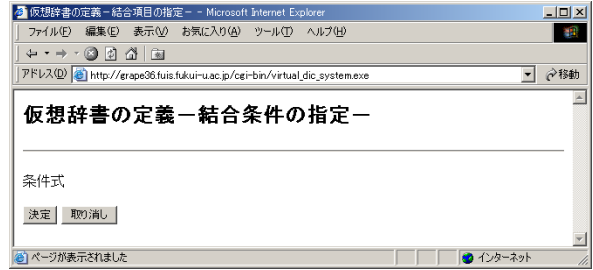


図 17 結合条件の指定画面③

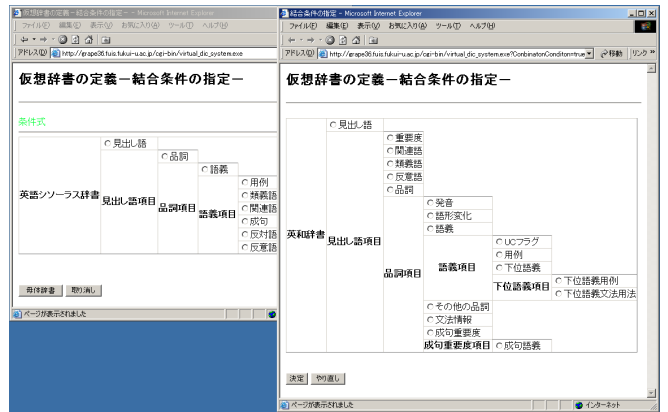


図 18 結合条件の指定画面④

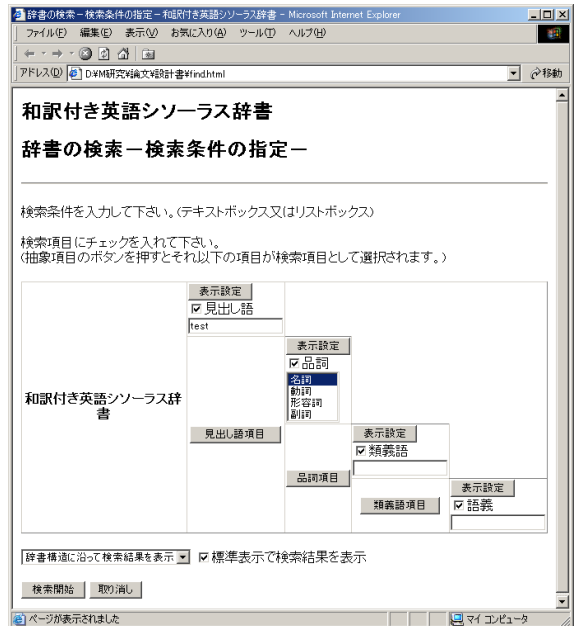


図 19 検索条件の入力画面

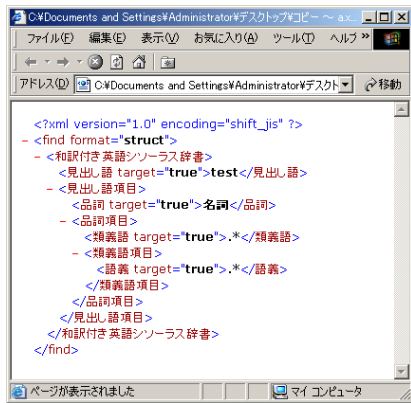


図 20 和訳付き英語シソーラス辞書 FIND 文

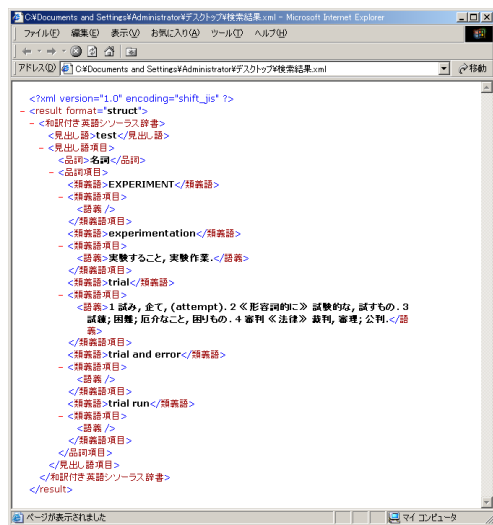


図 21 検索結果 XML

も、XSL を自動生成し、ブラウザに検索結果を表示する。この検索結果 XML と検索結果表示用 XSL は二次記憶上に格納され、次に本システムをユーザが使用する時にこの情報を元に一度検索した結果は仮想辞書サーバを通さず、結果を得ることが出来る。また、スタイルシートの変更、追加や削除なども行うことが出来る。

5. おわりに

現在、我々の研究室で開発中の仮想電子辞書システムにおける GUI について述べた。出力編成に関しては、まだ設計段階であるので、実装を行う必要がある。また、セキュリティに関して全く考えられていない。この部分も重要であるのでセキュリティを考慮した実装を行う必要がある。またエラー処理の強化も必要がある。ユーザにわかりやすい形でエラーを出すこともユーザインタフェースとして重要である。

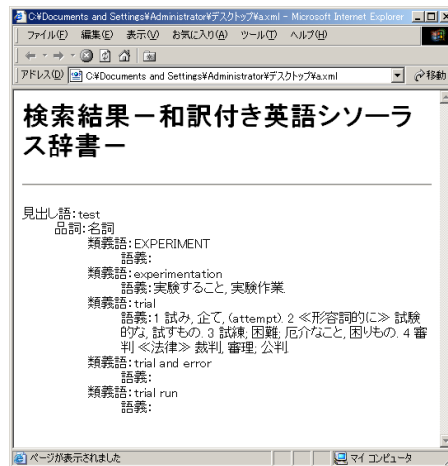


図 22 検索結果

出力編成を行う場合、図 23 の流れで行う。GUI では、各セレクトに配置されている表示設定ボタンは、初期状態は選択不可である。検索項目として選択された場合のみ選択可能状態になり、出力編成設定が可能である。これは、検索項目として選択されていないものは、検索結果に反映されないからである。表示設定ボタンを押すと、新しいウィンドウが現れる。ここでは、表示用の装飾設定と検索結果の情報を使った表示構造定義ができる。最後に検索開始ボタンを押す。

検索 GUI により、検索文の XML 表現が生成され、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検索文の XML 表現に直して、実体辞書サーバに検索させる。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て、検索 GUI へ返される(図 21)。この結果と出力編成情報を組み合わせて、XSL を自動生成する。そして、自動生成した XSL と検索結果 XML を組み合わせてブラウザに検索結果を表示する。出力編成設定を行わない場合

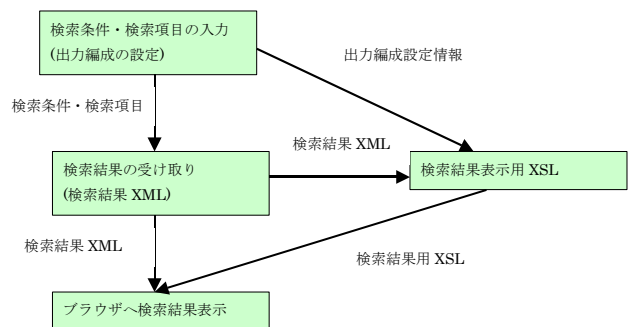


図 23 検索結果がブラウザに表示されるまでの流れ

文 献

- [1] W3C(XML), <http://www.w3c.org/>
- [2] EPWING コンソシアム, <http://www.epwing.or.jp/>
- [3] 近藤, 都司, 宝珍, 樋口: XML による仮想電子辞書の設計と構築, 電子情報通信学会第 14 回データ工学ワークショップ論文集, 論文番号 8A-4, 2003.