

M. Javaによるシソーラスブラウザの試作¹

The Prototype of Thesaurus Browser constructed by Java2/Swing

後藤 智範

神奈川大学 理学部 情報科学科

Thesauri have been widely used in bibliographic databases for 30 years. Recently, CD-ROMs of a variety of dictionaries with their GUIs are spreaded to current users. On the other hands, End users dose not use thesauri as for their bulky printed matter. The kind of software browsing graphically and managing thesauri does not appear in PC environment..

The browsing tool for lexical database with hierarchical structure has been developed using Java.. This paper describes the functions, the components, and examples of its usage. The problems of the browser and the functions to be extended are discussed.

1. はじめに

シソーラスが書誌データベースに利用されてから約30年が経過している。最近では、自然言語処理研究およびその応用ソフトウェアにも利用され、その応用範囲が拡大しつつある。現在、多くの辞書がCD-ROM化され、そのGUIソフトによって広く利用されているのに対し、検索ツールとしてのシソーラスは、未だに冊子体での利用が主であり、一般利用者には有効に活用されていない。シソーラスのグラフィカル表示についての研究は、1960年代のL. B. Doyleの“semantic road map”の研究^[1]に溯ることができるが、汎用性の高いシソーラスブラウザの普及には至っていない。

本研究室では、このような状況を改善するためにシソーラスおよび階層構造をもつ用語データに対する多面的なアクセスを可能とし、特定のシソーラスに限定されない汎用性の高いBrowserとその応用ソフトウェアの開発を行ってきた^{[2][3]}。開発最初期のブラウザはUNIXで標準的に使用されているXtoolkitを用いて試作を行った^[2]。1997年度から、Javaで再記述を行い、年々機能強化を行ってきた。本稿では、当ブラウザの最新バージョンの機能、利用実例、検討課題について報告する。

2. Browser の機能

用語間に階層構造を有するデータを多角的に、言い換えればその構造からアクセスするためのブラウザは、下記に挙げる機能を有することが必要とされる。

- (1) 見出し語の迅速かつ簡便な検索・表示。
- (2) 特定された見出し語の上位語ー下位語のグラフィカル表示。
- (3) 特定された見出し語の関連語の表示。
- (4) 履歴表示
- (5) 複数シソーラスの同時使用
- (6) マウス操作主体

¹本報告は下記の発表内容およびその後の発展も含めて書き直したものである。

後藤貴信、鈴木祐介、後藤智範. 階層構造をもつ用語データのためのBrowsing Tool. 情報知識学会第7回(1999年)研究報告会講演論文集, pp. 57-60(1999).

さらに、本年度新たに開発した下記に挙げる機能がある。

- (1) 上位語一下位語の縦型表示
- (2) 上位語一下位語の拡大・縮小表示

3. Browser の構成

本システムは、開発当初は Java(JDK1.0)およびその GUI コンポーネントである AWT(Abstract Windowing Toolkit)を用いて構築した。Java の version が上がる毎に対応するように部分的に書き換えてきた。現行ブラウザは JDK1.2 に対応しており、以下に示す本ブラウザのコンポーネントは、SWING を用いて実装した。JDK1.2 では、それ以前の GUI コンポーネントである AWT から SWING に変更され、これにより GUI コンポーネントの操作性、設計の自由度が大幅に向上した。現行の本ブラウザでは下記の 14 の SWING コンポーネントが用いられている。

DefaultListModel	Jbutton	Jdialog	JfileChooser
Jframe	Jlabel	Jlist	Jmenu
JmenuBar	JmenuItem	Jpanel	JPopupMenu
JscrollPane	Jviewport		

本ブラウザの動作の検証のためのシソーラスとして下記の 2 種類を用いた。

- (1) 1987年度版 INSPEC シソーラス 用語数: 約9000語(ディスクリプタ:約5500語)
- (2) 1987年度版 JICST シソーラス 用語数: 約 50000 語(ディスクリプタ:約 40000 語)

3.1 Browser の構成とその機能

本ブラウザは、4つの window コンポーネントから構成される。

- (1) メイン window
- (2) 用語リスト window
- (3) 用語木 window
- (4) 関連語 window

(1) メイン window

ブラウザの起動時に表示される window で、シソーラスの選択および、履歴ファイルの指定による過去の使用用語の履歴表示のために用いられる。図 1 にメイン window を示す。「シソーラス」ボタンを押すことにより登録されているシソーラスを選択する。

(2) 用語リスト window

見出し語を表示するための window で、JICST シソーラスを用いた実例を図 1 に示す。利用者が(1)で JICST シソーラスを選択すると、用語木 window は空白状態で表示される。次に、利用者が探したい用語の先頭文字を含む行(「0-9」～「わ」)のボタンをマウスクリックすると、当該行の文字(濁音などを含む)のポップアップメニューを表示する(この例では、「か」行)。利用者が先頭文字のボタンを押すと、頭文字から始まる見出し語が表示される。図 2 はこの状態を示しているが、ポップアップメニューは見出し語が表示されると瞬時に消える。右上の「UP」「DOWN」ボタンは、右側のスクロールと同等の機能を有し、「NEXT」ボタンは、以前に指定された用語を迅速に表示させるためのボタンである。

対象用語の総数が数万語以上になると、当該用語を見つけるためのスクロールが必要とされ、これが利用者に多大な手間と時間を課することが最初のbrowserの試作時に判明した。ポップアップメニューの導入により、見つけたい用語の範囲が限定されるためこの負担が軽減

されようになった。

用語リストwindowに表示される用語を指定し、上の「表示」ボタンを押すか、または右クリックのポップアップメニューにより、「NT-BT」、「RT」ボタンが現れる(図1では「感覚」を指定)。「NT-BT」ボタンが押されると、用語木windowが表示される。図3aにこの例を示す。

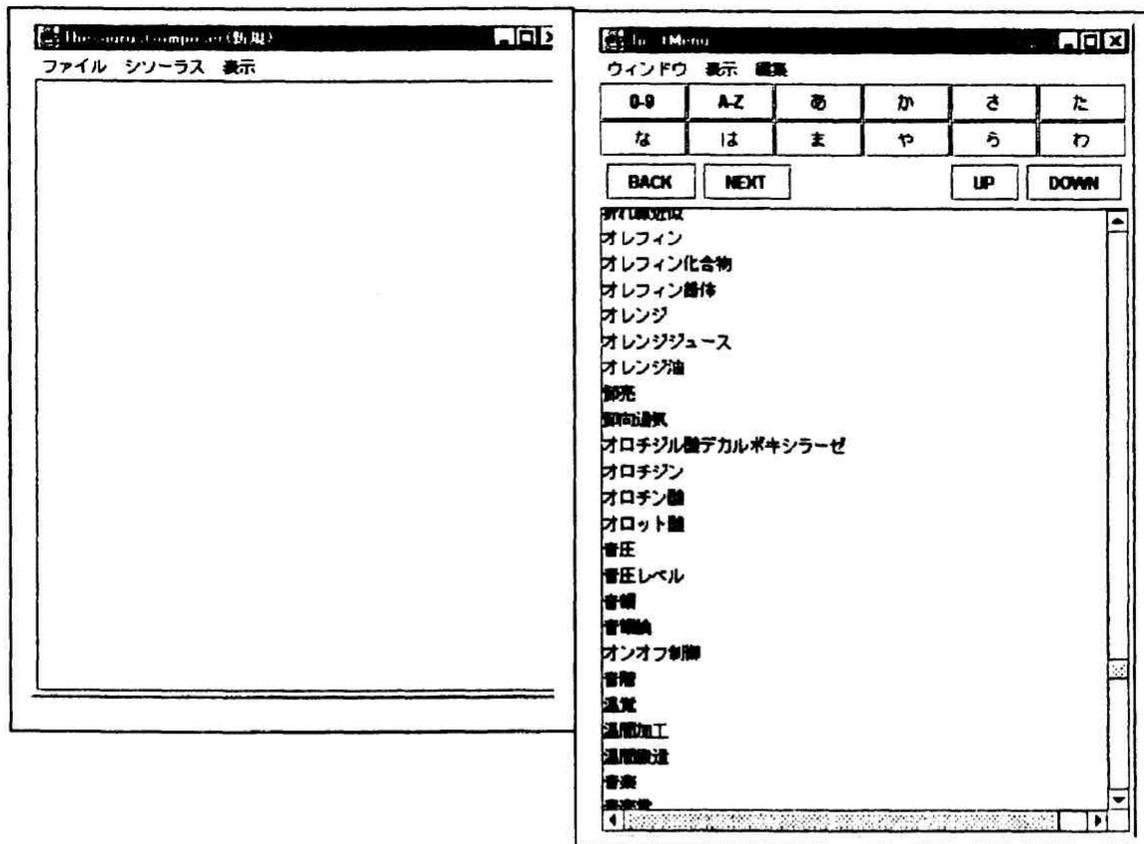


図1 メイン window

図2 用語リスト window

(3) 用語木 window

用語木windowは、(a)階層表示、(b)二次元カーソル(左上のボックス)、(c)構成用語リスト、(d)用語木情報、の4つのコンポーネントから構成されている。従来は、(a)だけであった。

シソーラスを構成する木の大きさ(用語数)、木の高さは様々であり、通常のディスプレイの解像度(1024×768~1600×1200)の範囲を超えるような木も数多く存在する。用語木を構成する個々の用語を閲覧するためには、縦横のスクロールバーによる操作が必須となるが、木が大き(100語以上)となると、この操作は利用者に多大な負荷を課す。(b)および(c)はこの操作を軽減するために導入したコンポーネントである。

二次元カーソルは、Xtoolkitにも含まれるコンポーネントであり、白枠が階層表示コンポーネントの描画面積を示し、黒い四角は、ディスプレイ上で実際に表示されている部分を示している。この黒い四角をマウスで動かすことにより、用語木の任意の場所を表示することができる。また、左に位置する用語windowの用語をクリックすると、階層表示コンポーネントはその用語がY軸上で中央にくるように、用語の表示色を赤に変更するように再描画し、同時に二次元カーソルも対応する位置に移動させる。図3bは図3aで、マウスを先頭の「感覚」から「温覚」に移動したときの状態を示している。「温覚」が赤に変わり、二次元カーソルの位置が3aとは異なった位置にあることがわかる。

この2つのコンポーネントの導入により、縦横のスクロールバーは不必要となり、また100語以上からなる大きな用語木についても、自在にその構成用語およびその上位語・下位語を知ることが可能となる。

上位一下位語、および関連語の表示は、用語が示されるどのwindowからでも表示できる。これは、階層構造コンポーネントに表示されている用語ノードからでも可能である。例えば図2において、「聴覚」、「視覚」のノードでマウスの右クリックを行えば、ポップアップwindowが表われ、関連語windowが表示される。

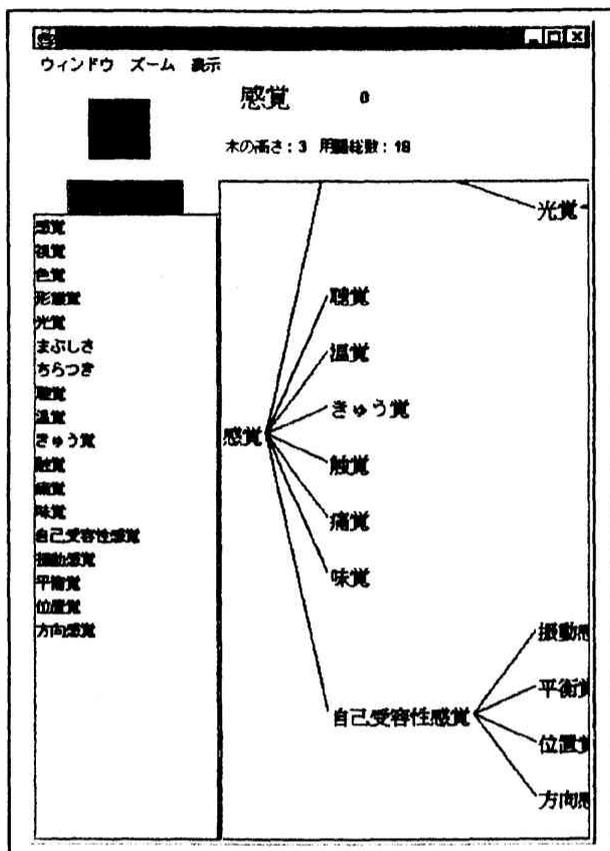


図3a 用語木ウインドウ

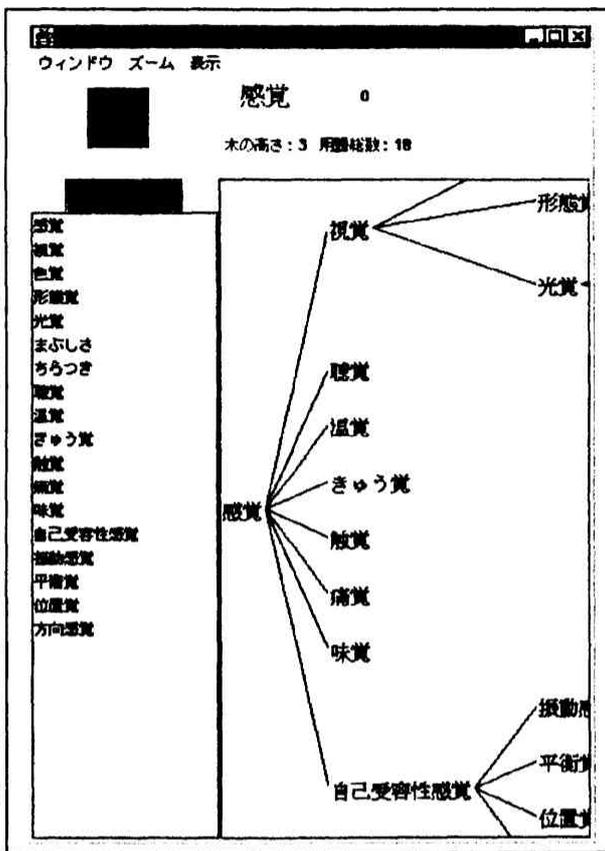


図3b 用語木ウインドウ

3.2 木構造表示の多様化

(1) 木構造の縦型表示

本年度の新たに開発した機能の1つに用語木 window での縦型表示機能がある。木構造の横型／縦型表示の切り替えは、ツールバー上の「表示ボタン」から縦型表示を選択することにより表示される。図4は、図3aの状態でのこの操作を行い表示された画面である。

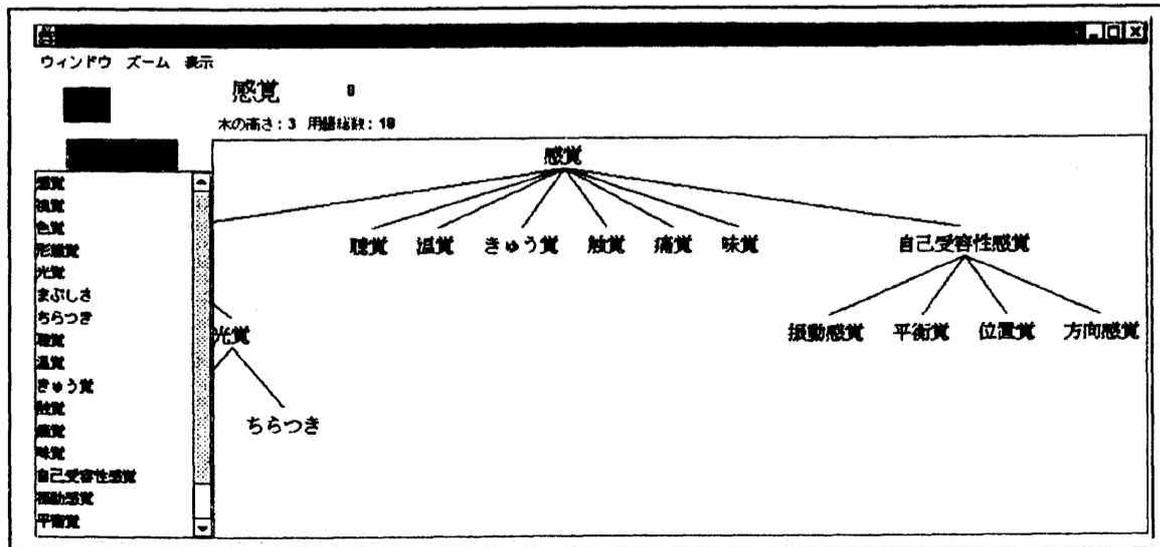


図4 用語木windowの縦型表示の例

(2) 木構造のズーム(拡大・縮小表示)機能

シソーラスには非常に多くのノード(用語)から構成される用語木があり、SXGA(1280×1024pixels)やUXGA(1600×1200pixels)程度の標準的なディスプレイ解像度では、用語windowを画面全体に拡大しても用語木全体を表示することができないことがある。この問題を改善するために、用語windowにおいて、マウス・イベントにより文字フォントサイズを動的に変更することにより、用語木に対する拡大、縮小機能を実現した。この機能は、ツールバー上の「ズーム」ボタンを押し、拡大/縮小を選択することにより実現される。図5は、図3aで縮小を選択して表示された画面を示している。図6は、図5で、拡大を3回続けて行った画面を示している。

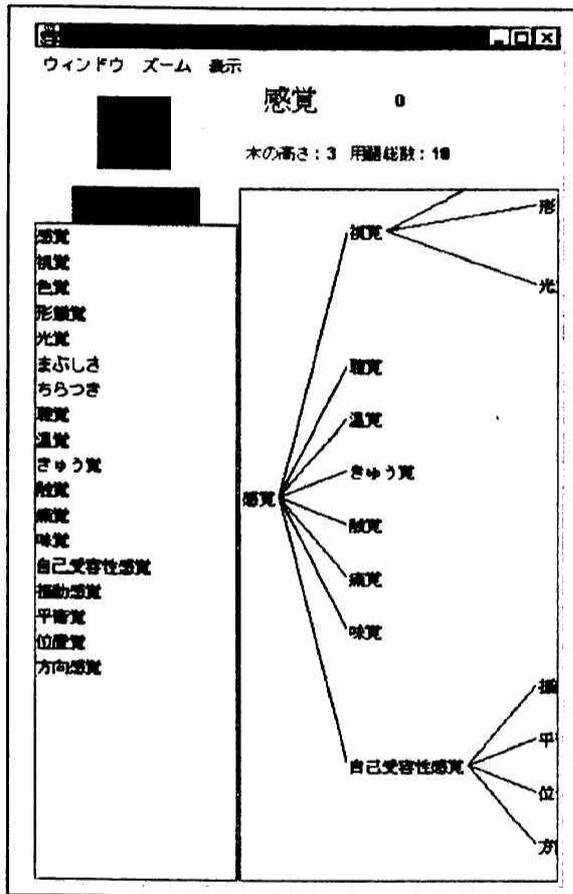


図5 ズーム機能—縮小

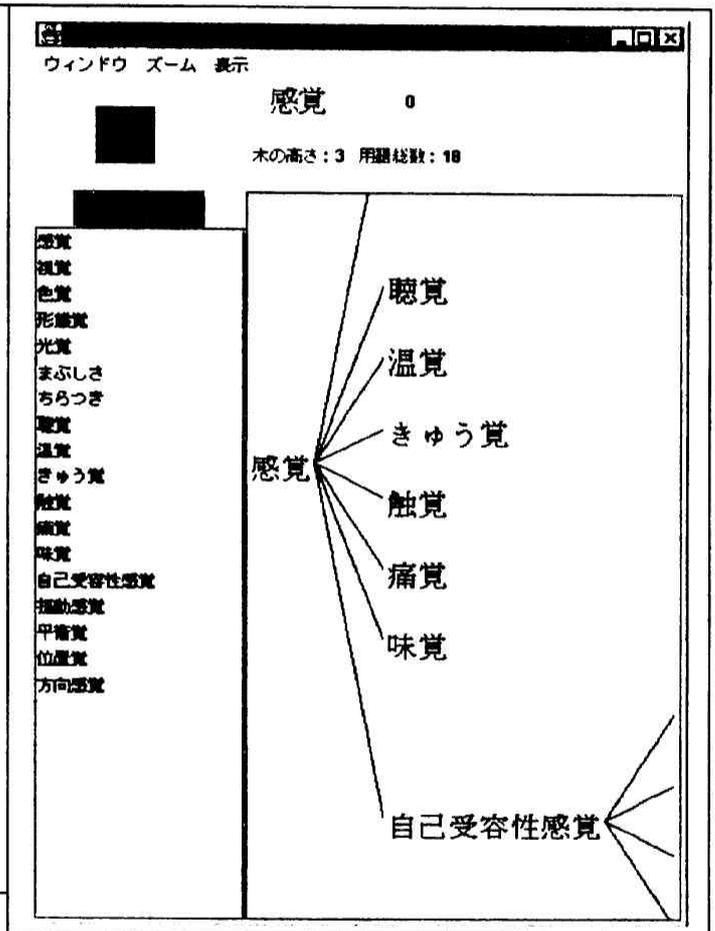


図6 ズーム機能—拡大

上述の木構造の縦型/横型選択表示およびズーム機能は、用語木windowが表示されている状態で、どの時点でも自由に組み合わせて使用することができる。

本ブラウザは、ブラウザが稼動するPC(windows95/98/NT), WS (UNIX)のメモリの容量が許す限り、1つのシソーラスに対して複数のwindowを表示することが可能であり、さらに同時に(1つのブラウザで)複数のシソーラスを利用することができる。図7は、INSPEC、JICSTの両シソーラスを同時に利用している例を示している。

複数のシソーラスの用語、木構造の比較などの目的に対しては、本ブラウザは極めて有効なツールとなる。

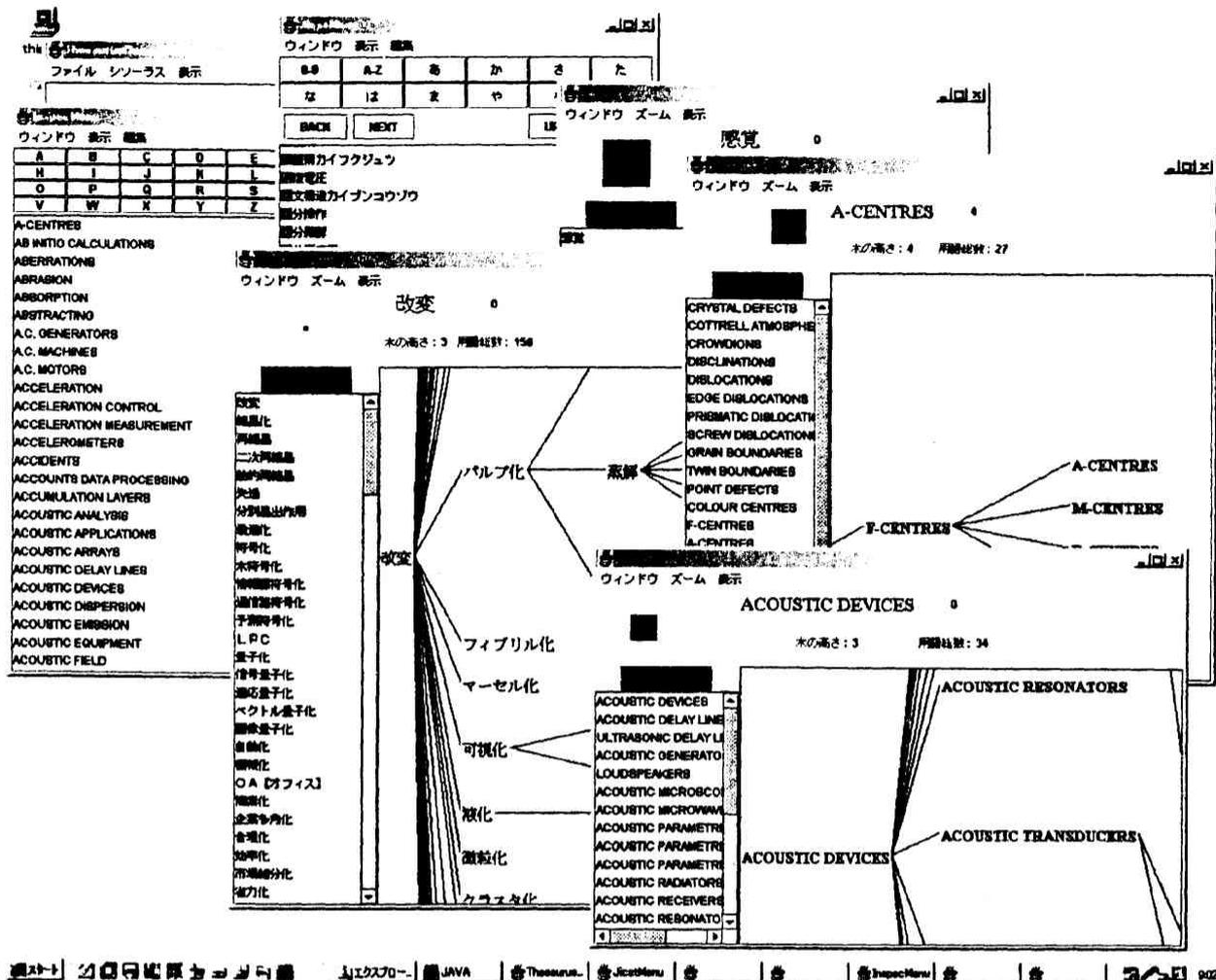


図7 複数シソーラスの利用例

4. 用語オブジェクトのデータ構造

4.1 シソーラスオブジェクトのデータ構造

Java1.2/Swing コンポーネントの導入は、本ブラウザにおいてロックアンドフィールドの向上だけにとどまらず、シソーラスオブジェクトのデータ構造の変更をもたらした。

本ブラウザでは、シソーラスに含まれる用語オブジェクトの格納にリストコンポーネントを用いている。Java1.1.Xでは、リストコンポーネントは文字列だけしか扱えなかったが、Swingではオブジェクトそのものが扱えるようになった。従来は、用語からその用語の下位語などの情報を保持しているインスタンスを得るために、データ構造としてハッシュ表を用いていた。上述のリストコンポーネントの拡張により、現行ブラウザリストコンポーネントに下位語などのデータを格納することで、ハッシュ表が不要となった。

4.2 シソーラスオブジェクトのデータ構造

従来の方式ではシソーラスファイルの読み込み時に全ての用語に対して座標計算用のクラスのインスタンスを生成していた。この方式では、利用者が表示しない用語についてはそのためのメモリの消費が無駄になっていた。この問題を改善するため、現行方式では、読み込み時には木構造の生成を行わず、用語木windowの起動により座標計算用のクラスのインスタンスを生成するような機構に変更した。

(1), (2)のデータ構造の変更によりメモリー使用量が大幅に軽減され、さらにブラウザの起動時間が短縮された。また、Java1.2/Swing に対応させた結果、ソースコードも減少した。表 1 にこれらについての、従来と現行のシソーラスの対比を示す。

表 1 : ブラウザーの性能

	ソースコード (行数)	メモリー使用量* (MB)	起動時間* (秒)
従来 version	3290	93	25.74
現行 version	2872	48	18.49

* Windows NT4.0 Workstation(CPU: Pentium Pro 200MHz, 主記憶: 128MB)

5. 検討項目と今後の課題

(1) シソーラスデータの記憶容量

表 1 から、メモリー使用量/起動時間は、相対的には大幅に改善されたが、起動時間は長く、主記憶容量も多く、アプリケーションソフトウェアとしては実用には達し得ない。起動時に読み込むシソーラスを変更し、主記憶容量を測定した結果を表 2 に示す。

表 2 シソーラスデータの容量

	起動時の主 記憶容量(MB)	ファイル 容量(MB)	レコード数
シソーラスなし	10		
INSPEC	15	0.5	5427
JICST	43	4.3	45137
INSPEC+ JICST	49	4.8	50564

この表から、シソーラス・オブジェクトの主記憶容量は、INSPEC、JICST でそれぞれ 5MB、33MB であることがわかる。表 2 から、主記憶、ディスクの 1 レコード当たりの容量を算出した結果を表 3 に示す。

表 3 シソーラスデータの 1 レコード当たりの容量(単位は Kbytes)

	主記憶	ディスク上
INSPEC	0.9	0.09
JICST	0.7	0.09

表 3 から、ディスク上のファイルと主記憶での容量が 2 つのシソーラスともに一語当たり約 10 倍に増加していることがわかる。本ブラウザでは、シソーラス読み込み時に、用語毎にインスタンスを生成しており、JICST シソーラスでは、約 45000 個のインスタンスが生成される。

Java では、あるオブジェクトのインスタンスについて継承機能により上位のクラスのオブ

ジェットのメソッドを継承するため、1つのインスタンスは様々なメソッドに対する参照情報を保持する。このため、メンバー変数の値の大きさ以上にインスタンスはの大きくなる。今後は、用語をオブジェクト(インスタンス)とするのではなく、シソーラス全体を1つのオブジェクトとすることにより記憶容量は大幅に削減できると考えられる。

(2) 今後の開発課題

シソーラス・ブラウザとしてさらにもつべき機能は以下であろう。

- (1) 見出し語を構成する単語(語基)からの参照
- (2) 用語木 window での構成用語のソート(用語およびその高さ)機能
- (3) ディスクリプター/非ディスクリプターの相互参照。

さらに、下記に挙げる編集機能を追加することを予定している。

- (1) 新規用語の登録・用語木の生成(シソーラスの新規作成)
- (2) 既存用語群の変更-用語木の編集(シソーラスの変更)
 - (a) ノードの変更、削除、(b) 部分木の削除、追加、(c) 部分木の合成

上記の機能を満たした時点において、本ブラウザを **freeware** として登録し、シソーラスの利用を広げることを計画している。

6. おわりに

近年、辞書・事典データの GUI^{[4][5]}、ハイパーテキスト化^{[6][7]}の研究が行われている。また、人工知能分野では、これらの研究に並行して、思考支援システム、ドメインオンロジー支援ツールの研究が行われている。いずれもその中心課題は用語-概念の表現、操作である。

本ブラウザは、可能性研究ではなく実用指向および汎用性を意図して開発したため、その機能をシソーラス、分類カテゴリーなどの階層構造をもつ用語データに限定しているが、本ブラウザのポップアップメニューに意味、訳語などの選択項目の追加、さらのオブジェクトの属性の拡張を行うことにより、汎用的な辞書 GUI に発展し、さらに推論機能を付加することにより思考支援システムに接近するものと考えている。

参考文献

- [1] Doyle, L.B. "Indexing and abstracting by association" *American Documentation*, vol.13, no.4, pp.376-390(1962).
- [2] 下村 央人 他.『汎用シソーラスブラウザの試作』.第 33 回情報科学技術研究集会発表論文集,pp.99-105(1997).
- [3] 下村 央人 他.『JAVA による索引支援ブラウザの試作 2』.第 35 回情報科学技術研究集会予稿集,pp.133-137(1998).
- [4] 藤澤浩道 他. 概念ブラウザと個人情報環境. 情報処理学会研究報告. No.53, 情報メディア-7, pp.41-48(1992).
- [5] 都司達夫 他. 仮想電子辞書の設計. 情報処理学会研究報告. No.106, 情報学基礎-48, pp.27-34(1997).
- [6] 黒橋禎夫 他. 専門用語辞典の自動的ハイパーテキスト化の方法. 人工知能学会誌 Vol.7, No.2, pp.336-345(1992).
- [7] 内藤広志 他.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. vol.34, pp.320-330(1993).