ソフトウェア再利用による語彙調査用ツールの開発

Development of a Software Tool for Japanese Language Analysis by reusing Software from a Project

佐野洋　　幸松英恵
SANO Hiroshi　YUKIMATSU Hanae

東京外国語大学 外国語学部　　東京外国語大学 大学院地域文化研究科
Tokyo University of Foreign Studies, Faculty of Foreign Studies

1. はじめに

1.1 言語を分析する知識

言語の分析過程や方法論を手続き化したソフトウェアも言語知識データベースの一つと見なすことができる。それは，研究成果を生みだした道具としてのソフトウェアには，研究者の長年にわたる研究目的に従った創意，工夫そして思考過程が部分的に手続き化されているからである。しかし，ソフトウェアとしての言語知識データは，最終成果の論文，辞書や報告書の陰に隠れて注目されることが少なかったように思われる。

FD，CD-ROM などデータやソフトウェアの記録媒体を書籍に添付できるようになって，研究成果を産出する過程で使われたソフトウェアが流通するようになった。インターネットを使ったウェブページの開設が身近になった現在，例えば，ダウンロードを通じたデータやソフトウェアの頒布と，そのソフトウェアを産出したプロセスの記録の重要性の認識が高まっている。

1.2 ソフトウェアの保守

ところで，一般にソフトウェアは，開発されてからの時間の経過とともに，実行環境であるハードウェアやオペレーティングシステム(OS)が変化する。そのために新しいプラットホームへプログラムを移行する(プログラムを書き換える)必要性が生じる。こうした，ソフトウェアをデータ環境の変化や処理環境の変化に適応させる作業は，適応保守と呼ばれる[1]。新規開発よりもコストを要することもあるという。

本稿で示す日本語の調査・研究のためのソフトウェアは，既存ソフトウェアの適応保守の工程を経て開発されたものである。適用保守の対象のソフトウェアは，『パソコンによる日本語研究法入門—語彙と文字—』(中野洋著，笠間書院，1996 年)に掲載の日本語研究のためのソフトウェア群(MCL)である。

筆者は，AWK 言語で記述された日本語研究用ソフトウェア群(MCL)を取り上げ，処理環境の変化に適用させる保守を実施し，Perl 言語でプログラムを再実装した(表 1を参照)。このソフトウェアを CLTOOL と呼んでいる。本稿では，以下，CLTOOL について概略を説明し，主なプログラムの実行の様子を示す。

2. Perl 言語によるプログラムの実装

2.1 CLTOOL

MCL[2]を参照し，その AWK プログラムを分析し，その設計復元を通じて問題領域に関する知識(日本語研究の方法論についてのプロセス知識)を抽出したうえで，Perl プログラムにした。プログラムを保守しやすい構造に変え，現代的なユーザーインタフェース部分を新規に作成した。さらに今後の保守のために，ソフトウェアに表現されたことばの分析過程(日本語研究の方法論についてのプロセス知識)を再文書化した。

表 1. ソフトウェア諸表

	文献[2]	本稿
開発言語	AWK 言語	Perl 言語
OS	MS-DOS	Windows XP/ 2000/98/Me/95
インタフェース	CUI	GUI
実行環境	JGAWK	JPerl，Perl/Tk

2.2 設計復元と作業項目

作業項目を以下に示す.

処理ロジック：基本的に処理ロジックの変更はない。ただし，(1) ソフトウェア・インタフェースとして Perl/Tk を利用した GUI を採用し，幾つかの手続きを GUI 上での対話操作を通じて処理を行うようにした。バッチ処理プログラムをなくした。(2) Perl 言語は AWK 言語より処理手続きの記述力が高いことから，一部の処理ロジックの変更を行った。コード上は，AWK 言語は Perl 言語のほぼサブセットであることからプログラム上の記述差は大きいものではない。(3) 文字，語彙調査だけではなく，複文，連文調査にも活用できるように機能拡張した。例えば，各プログラムの処理結果のデータ書式を csv 形式にした。また，kwic は，文単位の文脈指定が可能である。日本語では文特徴が文末述語に現れることから，文字列の逆順整列の機能を kwic と連携させ，分析能力を向上させた。

プラットホームとインタフェース：現代的な GUI 環境を持つ OS で使えること，想定利用者である人文

系の言語研究者や学生等に利用し易いことなどの観点から Windows R 98 及びそれ以降の版に対応できるようにした。インタフェースは，想定利用者とツールを結びつける手段である。利用者とツールの機能－ことばの調査，研究－を効果的に発揮させるために重要な役割を演じている。従って，インタフェースは，利用者がツールを使って作業をする際の要求に適合するべきだ。本ツールのインタフェース設計では，本学の(日本語研究を専門とする)大学院生に使用させてモニター評価を実施した。使用感を分析し，その結果をインタフェースデザインに反映した。

利用のモデル：CLTOOL では，操作の視覚化と処理結果の視覚化を分けている。GUI を通じた操作の視覚化によって利用者は，試行錯誤的に言語分析を行うことができる。処理結果は，CSV 形式を採用し，エクセルなど表計算ソフトウェアで統計分析やグラフ化などの視覚化が可能にしている。すなわち，本ツールは，閉じた状態で利用するのではなく，他のアプリケーションと組み合わせて利用することを前提に設計されている。

3. CLTOOL の概略

3.1 ソフトウェア一覧

表 2に主なソフトウェアの一覧を挙げる。ソフトウェア名は，MCL[2]の機能名と可能な限り対応させている。

表 2. ソフトウェア一覧

ソフトウェア	機　能
cat, wc, word, xdump	テキストを表示する。cat はテキストデータを表示し，複数のテキストを指定することで，それらを一つにまとめる。wc は文字数やレコード数を集計する。word は，単語ごと，字種ごとにテキストを分割する。単語の分割手続きは MCL の対応ソフトウェアに準じる。xdump はテキストを文字コードで表示する。
moji	テキスト内の文字を文字種ごとに分けてその文字数を計測する。計測の手続きは MCL の対応ソフトウェアに準じる。
record	句点，読点などを区切り子を指定してレコードデータを作成する。
grep	特定の文字列を指定し，その文字列部分を前後のテキスト(文脈)付きで抽出する。
kwic, kwic2	特定の文字列を指定し，その文字列部分を前後のテキスト(文脈)付きで抽出する。指定された文字列や文脈文字列の整列機能を持つ。
filter	会話文の抽出，全角から半角への変換，ルビ削除といったテキスト整形を行う。
csvfilter	CSV 形式のデータ処理を行う。特定の列を指定したキーワード検索や特定項目の分割と統合，逆順整列機能を持つ。一般の表計算ソフトウェアにない機能を提供する。kwic, kwic2 と連携させて利用することを前提とする。

CLTOOL では，MCL[2]の中の対照言語調査用プログラムならびに一貫処理プログラムなどは実装していない。

3.2 インタフェースデザインと機能概要

CLTOOL の基本操作を担うインタフェースデザインを説明する。図 1に cat プログラムのウィンドウを示す。

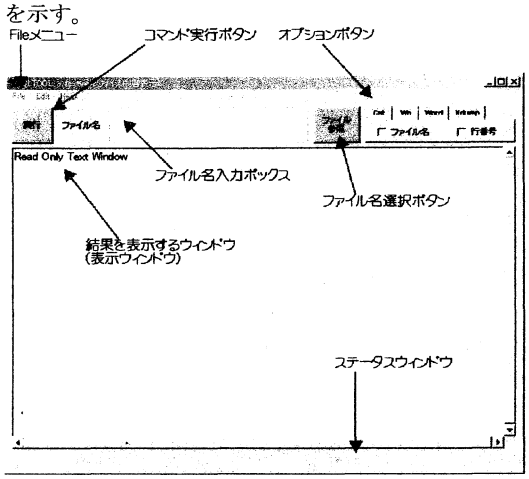


図 1. ソフトウェア外観

File メニュー：File メニューには，[OpenStream]，[SaveFile]，[Exit]の項目がある。プログラムの終了は，File メニューの[Exit]を選択するか，ウィンドウ右上端のボタンを押す。一般的なウィンドウズ・アプリケーションの操作に模して，利用者の操作負担を軽減している。[OpenStream]を選択すると，ファイル名入力ボックスに instream というファイル名が表示される。このファイルには，CLTOOL のどれかのプログラムで直前に分析したデータの結果が記録されている。このデータが現在のプログラムの入力データになる。MCL[2]でも，バッチプログラムで採用されていたパイプ処理に似せた概念を視覚的に提供している。[SaveFile]を選択することで，処理結果を任意の名前のファイルに記録することができる。

ファイル名入力ボックス：分析対象のファイル名を入力する。一般には，File メニューに[Open]を設けず，分析対象ファイルを明示的に意識させるインタフェースとした。

ファイル参照ボタン：ファイル名入力ボックスからキーボード入力は，実質的に行われないし期待していない。通常は，ファイル参照ボタンを使ってファイル名を入力する。

実行ボタン：実行ボタンをクリックすることで，プログラムが実行される。ほとんどのプログラムの実

行の結果は「Read Only Text Window」部分に表示される。処理結果のモニター画面の役目を果たす。処理結果は，同時に CLTOOL フォルダ内にある outstream ファイルにも記録される。

オプションボタン：プログラムで分析を行う際に，オプションボタンを使って付加的な条件を指定する。付加的な条件は，プログラムの機能によって違い，オプションの指定方法に応じたインタフェースが用意される。

Read Only Text Window：プログラムの実行結果が表示される，このウィンドウはスクロールウィンドウになっている。このウィンドウ上に表示されるテキストはカット&ペーストなどのテキスト編集を禁止している。試行錯誤な分析を含め，結果を確認するためだけの表示機能と位置づけている。編集可能なデータは，全て outstream ファイルに自動的に保存される。

ステータスウィンドウ：ウィンドウの最下層には，プログラムの実行状態が示される。プログラムの実行状態を確認することができる。

3.3 stream ファイル

CLTOOL を構成する各種のプログラムは，実行のたびにその結果を outstream という名前のファイルに記録する。File メニューの[OpenStream]を選択すると，その時点での outstream ファイルの内容が instream ファイルに複写され，その instream ファイルがファイル名ボックスに設定される。そして，分析対象のデータとして利用される。

MCL[2]で，バッチプログラムで採用されていたパイプ処理に似せた概念を提供する。ファイル拡張子はないので stream ファイルをクリックしても特定のアプリケーションは起動しない。現在，インタフェースのデザインを含めて，分析データと関連アプリケーションが利用者に明示できる概念化の方式を検討している。

4. CLTOOL の利用

主なプログラムの実行の様子を示す。

4.1 moji

moji は，テキスト内の文字を文字種ごとに分けてその文字数を計測する。図 2には，moji の実行の様子を示す。このプログラムは，文字分布のデータを画面上でグラフ表示する。同時にそのデータは，CSV 形式でもファイル出力されて，エクセルなどの表計算ソフトウェアを使って，データ処理を行うことができる(図 3を参照)。1 文字毎の頻度データも同様の形式でファイル出力されるので，エクセルの統計処理機能やグラフ化機能を使うことで，多視点の分析が可能になる。

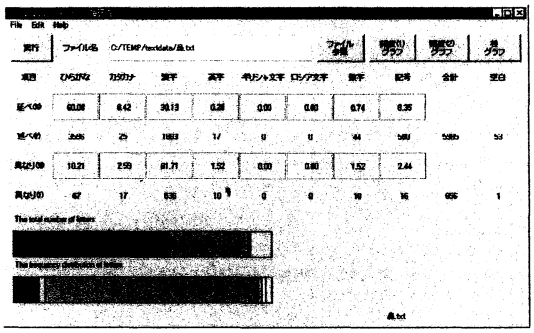
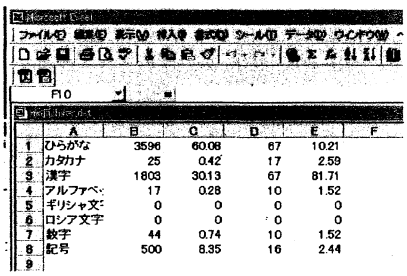


図 2. moji の実行の様子



	A	B	C	D	E	F
1	ひらがな	3596	60.08	67	10.21	
2	カタカナ	25	0.42	17	2.59	
3	漢字	1803	30.13	67	81.71	
4	アルファベ	17	0.28	10	1.52	
5	ギリシャ文	0	0	0	0	
6	ロシア文字	0	0	0	0	
7	数字	44	0.74	10	1.52	
8	記号	500	8.35	16	2.44	
9						

図 3. 文字分布データ

4.2 grep/kwic

grep は，特定の文字列を指定し，その文字列部分を前後のテキスト(文脈)付きで抽出する。kwic は，特定の文字列を指定し，その文字列部分を前後のテキスト(文脈)付きで抽出する。指定された文字列や文脈文字列の整列機能を持つ。図 4に実行の様子を示す。いずれも一般的なソフトウェアと同等機能である。

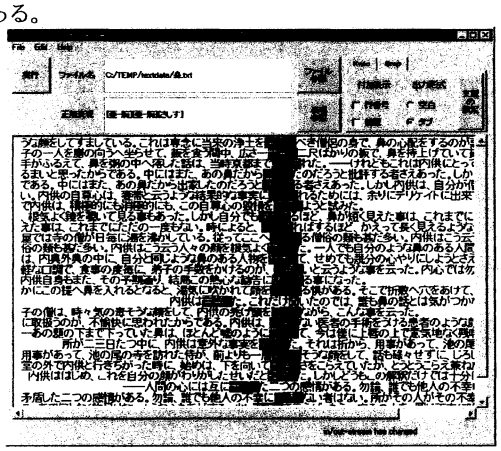


図 4. kwic/grep の実行の様子

人文系の言語研究者や学生等が想定利用者である。

これら想定利用者の利用し易さを考え，図 5に示すインタフェースを用意した。数値の設定と変量を直感的に把握と指示ができるスライダーを文脈数(と文数)の設定に用いている。正規表現は，正規表現の選択パネルからボタンクリックを通じて入力できる。(1) 入力の手間を軽減すること，(2) 照合の適用例を予め示すことで，正規表現の学習を支援すること等の便宜を図っている。さらに，利用者による正規表現の登録機能の充実を予定している。

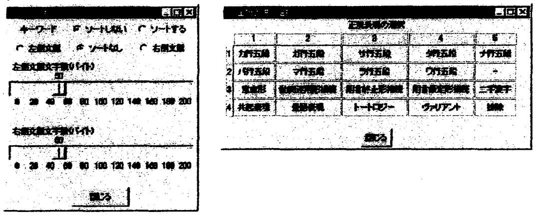


図 5. 文脈数の設定画面と正規表現の設定画面

4.3 csvfilter

csvfilter は kwic と連携させて利用するレコードデータの整形プログラムである。次の機能を持つ。

1. kwic キーワードの分割：CSV データの第 2 項目を指定文字列で分割する。
2. 項目統合：CSV データの隣り合う項目データを 1 つの項目にまとめる。
3. 項目内文字検索：CSV データの項目を指定し，その項目内で文字列検索をする。
4. 項目内分離：CSV データの項目を指定し，指定文字列を別の項目に分離す
5. 逆順整列：CSV 形式の項目を指定し，逆順整列する。

例えば，「伝聞」を表す「そうだ」を kwic で抽出する場合，正規表現として「[うくすつぬむるぐいただ]そうだ」を入力する。分析対象の文字列「そうだ」に前接する語尾部分(うくすつぬむるぐぶいただ)までがキーワードに含まれている。

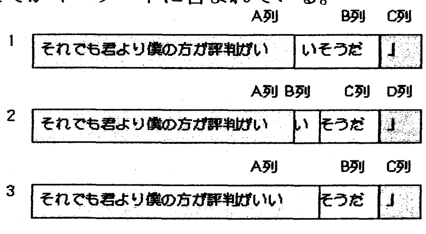


図 6. 「～そうだ」の整形の様子

kwic の照合結果の B 列を対象に「そうだ」とそれ以外で分離し，A 列と B 列の項目統合を行うことで，「そうだ」をキーワードにした検索と同等となる。A 列に対して逆順整列を行うと「そうだ」に前接する語尾の順で並べることができる。

キーワードの前後文脈の整列だけではなく，キーワードの後ろ文脈を基準に，csvfilter を使ってキーワード部分の逆順整列を行うと図 7に示すように複文の調査などに応用することができる。

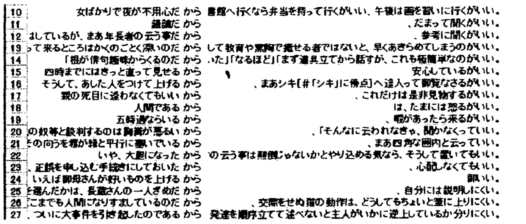


図 7. 「から」の後件を中心にした逆順整列

5. おわりに

本稿は，日本語研究の方法論についてのプロセス知識としてのソフトウェアの保全と再利用について述べた．

5.1 保全と再利用

筆者は，言語研究を目的として作成されたプログラムには，ことばの研究プロセスに関する知識が含まれることに着目した．プログラム保守の枠組みを応用し，設計復元を通じて，これらの知識を抽出するだけでなく，ドキュメンテーションの考え方を利用して，ことばの分析過程を再文書化している。

筆者は，この文書を言語研究プロセスドキュメントと呼んでいる．言語分析の方法論や技術論など，一部は研究成果報告や研究資料に含まれることがあるが，処理プログラムの細かな開発手順や，暗黙に仮定されている分析対象の言語特徴など，従来は記録されることが少なかった．本稿で示したプログラムもいずれ他のプログラム言語を使い適応保守の枠組みで再開発しなければならない。言語データといった知識と同時に，ことばの研究プロセスの知識も継承していくことが重要である．

参考文献

[1]「保守とエンジニアリング(738p～746p)」,「開発管理(759p～772p)」, 情報処理ハンドブック，情報処理学会編,1995.

[2] 中野洋,『パソコンによる日本語研究法入門』,笠間書院,1996.

[3] 吉川弘之監修,『技術知の位相―プロセス知の観点から』,東京大学出版会,1997.

[4] トニー・グラハム著,関口正裕監修,『Unicode 標準入門』,翔泳社,2001.

[5] 中島靖,『日本語 TEXT 加工実践ガイドブック』,情報管理,1997.

[6] 佐野洋,「WindowsPC による日本語研究」,共立出版,2003(出版予定).