N-9

ロマンス諸語のための言語分析ツールの開発

大家正登，山田純子，佐野 洋

1. はじめに

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(adaptive maintenance)と呼ばれる[1,3].

本稿は，適応保守の考え方を基に，ロマンス諸語のための言語分析ツールを開発したので報告する．AWK 言語で記述されたスペイン語研究用プログラムを取り上げ，処理環境の変化に適用させる保守を実施し，Perl 言語で言語分析ツールを再実装した．設計復元を通じて問題領域に関する知識(言語研究プロセスについて言語知識)を復元し文書化し，該当個所をモジュール化した．プログラムを保守しやすい構造に変え，GUI を新規に作成した．同時に，処理過程と言語知識を区分した結果を利用し，ポルトガル語分析ができるよう機能拡張を実施した．今後，フランス語分析やイタリア語分析の機能を実装する予定である．

1.1 言語知識データの保全の必要性

言語研究プロセスなどの言語知識データの保全(共有化と再利用)は，次の理由から重要である．

(1) 再構築が困難である．ことばの分析は，研究者への依存度が極めて強く創造的な活動である．おなじ品質の言語知識データを再構築することは困難である．

(2) 言語データの収集や作成にかかる人件費コストが大きい．言語データの収集方法の考案，実際の収集や作成への人件費投入コストの割合が大きいため，作業を繰り返すのは無駄である．

(3) 知識資産である．その時代にのみ収集することのできる言語データがある．ことばという社会システムは時間に沿って変化していると言われ，そのため時間を溯って言語データを再現することができない．

筆者等は，言語研究を目的として作成されたプログラムには，言語研究プロセスの言語知識データ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，モジュール化やドキュメンテーションの考え方を利用して，保守が可能なプログラムに変えることで(1),(2)の問題点を解消しようとするものである．

1.2 保全のポイント

保全の技術面に関係するのポイントを挙げる．なお，表現データ上では著作権等の法的な問題も大きいが本稿の議論の範囲でない．

1. 言語知識データの流通性と閲覧の容易性を確保した技術枠組みに沿ってソフトウェアを構築するべきである．
2. データの意味内容の理解が容易であるべきである．言語データの公開や提供は単なるデータソースの提供ではなく，ドキュメンテーション技術も含める．
3. ことばの分析過程のソフトウェア化に対する品質管理の考え方の枠組みを決めるべきである．とくに，多言語対照研究に応用するとき，言語知識データの品質管理は重要である．
4. 維持と保全に対するコスト投下と技術導入を考慮するべきである．データ収集と蓄積だけでなく，データ維持と保全にかかる費用と応用技術を明らかにする．

東京外国語大学 外国語学部

2. スペイン語用 KWIC

2.1 AWK 言語による実装

宮本(神戸市外国語大学)[5]によって開発されたスペイン語用 KWIC は，スペイン語コーパスを対象に，その構文特徴を調査するためのプログラムである．AWK 言語で記述されている．このプログラムは，検索対象のスペイン語の語句(キーワード)を中心に，その語句の左右の文脈を表示する．ただ，プログラムが扱う問題領域がスペイン語研究であることから，日本語用あるいは英語用の KWIC プログラムと違う属性を持つ．

スペイン語は，いわゆるロマンス諸語に属し，動詞の活用の豊富さに 1 つの特徴がある．すなわち，1 つの動詞は，不定詞形，現在分詞形，過去分詞形，直接法現在形，直説法点過去形，直説法線過去形，直説法未来形，直説法過去未来形，接続法現在形，接続法過去形(-ra, -se)，接続法過去完了形(-ra, -se)，直説法現在完了形，直説法過去完了形，直説法直前過去形，直説法未来完了形，直説法過去未来完了，接続法現在完了形，接続法過去完了形(-ra,-se,-re)，不定詞の複合形，現在分詞の複合形，命令法の各活用形を持つ．これらの活用形は人称と数に反応するので，例えば，規則動詞 hablar は 118 の活用形態を持つことになる．

KWIC プログラムは，検索語句に述語を含むとき，その述語の展開リストを生成する．その生成されたリストを用いて語句検索が行われる．従って，このプログラムの設計情報のうち重要なポイントは，ソースファイルに含まれるデータ項目になる．特に，語彙検索の機能部分のデータ項目の依存関係を辿ることで，特定のデータの定義箇所や生成源が分かり，データ項目に関係する分岐条件等，その使われ方と変数領域等の記憶スペースの使われ方が分かる．

2.2 Perl によるプログラムの実装

筆者等は，再構造化(restructuring)を実施し，AWK 言語で記述されたスペイン語用 KWIC プログラムを Perl 言語で書き直した．この作業は，プログラムの持つ機能の抽

象度を変えることなく，AWK 言語記述から Perl 言語記述に変形することである．作業内容を表 1に示す．

表 1. 言語分析ツールの開発項目(1)

処理ロジック	言語研究プロセスに変更はないので，基本的に処理ロジックに変更はない．ただし，スペイン語固有の属性に対応するデータ項目については，モジュール化を行い一部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記述差は小さい．
処理系	Perl(5.6.1)
プラットホーム	現代的な GUI 環境を持つ OS で使えること，人文系の言語研究者や学生等に利用し易いことの点から Windows 95(R)(あるいはそれ以降の版)．
文書化	処理ロジックに関する文書だけでなく，利用者マニュアルと保守マニュアルを整備した．

保守しやすいプログラムの要件は理解しやすさにあるという[3]．主な再構造化のポイントは，(1) スペイン語固有の属性に対応するコード部分をモジュール化した，(2) 変数名，手続き名は機能を表現する名前に変更し，十分な量のコメントを挿入した，(3) プログラムの目的，入出力データ，使用例，動作状態の確認の仕方など，効率的かつ効果的に保守を行えるように再文書化(re-documentation)に注力した，などである．

筆者等は，(3)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，とくに多言語では，ほとんど記録されることがなかった．

2.3 ポルトガル語への対応

ポルトガル語も動詞の活用が豊富である．すなわち，1つの動詞は，不定詞形，現在分詞形，過去分詞形，直説法現在形，直説法現在完了形，直説法不完全過去形，直説法過去完了形，直説法完全過去形，直説法大過去形，直説法未来形，直説法未来完了形，直説法過去未来形，直接法過去未来完了形，接続法現在形，接続法現在完了形，接続法過去形，接続法過去完了形，接続法未来形，接続法未来完了形，命令法形，人称不定詞形，完了不定詞形の各活用形を持つ．これらの活用形は人称と数に反応するので，例えば，規則動詞 passar は 113 の活用形態を持つことになる．

モジュール化したデータ項目の処理部分をポルトガル語用に変更することでポルトガル語へ対応させた．同時に保守マニュアルに追加モジュールの情報の追加記述を行った．

なお，モジュール部分は，今後，フランス語やイタリア語にも対応させる予定である．

3. 機能拡張

3.1 多言語(ロマンス諸語)対応

処理対象の文字コードを UNICODE とすることで，多言語分析に対応させた(表 2を参照)．

表 2. 言語分析ツールの開発項目(2)

機能拡張	処理対象の文字を ASCII コードから UNICODE へ拡張した．GUI インタフェースを新規に設計し開発した．
処理系	Perl/Tk, Java.
プラットホーム	2.2節の理由に加えて，UNICODE(UTF-8)データを扱うことの点から Windows 2000(R)(あるいはそれ以降の版)．

3.2 インタフェース

表 2に示すように GUI インタフェースを新規に設計し開発した．この保守作業は，利用者要求(コマンドラインからのプログラムの実行操作ではなく，現代的な OS が提供する GUI 操作とシームレスな実行操作を実現すること)に応えるものである．以下の 2 つの方法を用いた．

3.2.1 Perl/Tk による開発

Perl/Tk は，Perl で GUI プログラミングを実現するモジュールである．現版の Perl/Tk は，UNICODE を直接入出力することができない．そのため，パラメーター指定を含めた実行操作の GUI 化と，キーワード指定と処理結果を指定するファイル参照の GUI 化を行った．

3.2.2 Java を使った開発

Java は，ネイティブコードで UNICODE をサポートする．ウィンドウクラスを使い，UNICODE でのキーワードの指定や処理結果の表示を行う GUI 部分を作成した．Java から Perl プログラムをコールする．

いずれの変更も，利用者の満足度高めるための修正作業であるが，次の，・プログラムの変更を単純化する，・今後のソフトウェア保守に必要なコストを低減する，という観点から評価を行っている．

4. おわりに

本稿では，言語研究プロセスを含め，言語知識データの流通性や保全性が重要であることを指摘した．

優れた言語研究者が優れた言語知識データを作る．言語データだけでなく言語研究プロセスも言語知識データである．こうした学術資産を作り始めるスタートポイントは人材である．しかしながら，人材を育てるために教育し，研究環境を整え，報酬を十分なものにし，行動の動機を与えることは容易なことでない．従って，各種の記録媒体や記録手段で記された言語知識データ(言語資源データとプログラム)は，その活用のため維持と管理を行い，何年にもわたり保全して資産価値を守る必要がある．

参考文献

[1] 「保守とエンジニアリング」,情報処理ハンドブック,738p～746p,情報処理学会編,1995.
[2] 「開発管理」,情報処理ハンドブック,759p～772p,情報処理学会編,1995.
[3] Carma McClure 著,ベスト CASE 研究グループ訳「ソフトウェア開発と保守の戦略」,共立出版株式会社,1993.
[4] 斎藤俊雄,中村純作,赤野一郎「英語コーパス言語学」,研究者出版株式会社,1998.
[5] 宮本正美：「El habla de la Ciudad de Madrid」の動詞句コンコーダンス(acabar-permitirse)」,神戸市外国語大学外国学研究所，1993.