

ハードウェアの特徴を活かしつつ移植性のよい
 応用ソフトウェアを開発するための一方法

5L-8

— 日本語文章推敲支援ツール『推敲』の移植経験から —

倉田昌典 菅沼明 牛島和夫
 (九州大学工学部)

1. はじめに

パーソナルコンピュータは、ユーザインタフェース設計の自由度が大きい。そのため、パソコン上の応用ソフトウェアには、ユーザインタフェースの優れたものをよくみかける。しかしながら、その優れたインタフェースは、ハードウェアに依存することによって実現されている場合が多い。このことは、応用ソフトウェアをさまざまなパソコンに移植する際の障害となっている。一方、移植性を考慮して応用ソフトウェアを開発しようとする、ユーザインタフェース設計の自由度が小さくなってしまふ。

われわれの研究室でパソコン上に開発している日本語文章推敲支援ツール『推敲』は、日本語文章ファイルを解析して、推敲に役立つ情報を提供する応用ソフトウェアである。このたびこの『推敲』を、いくつかのパソコン上へ移植した。その過程は、前述の問題点について考察するよい機会となった。

本論文では、『推敲』移植の際に生じた問題点を挙げ、その経験からわれわれが考えた、ハードウェアの特徴を活かしつつ移植性のよい応用ソフトウェアを開発する一方法について述べる。

2. パソコン版『推敲』

2.1 開発動機

われわれは、機械可読の日本語文章から何とか推敲に役立つ情報を取り出してみたいという動機で、1983年、大型計算機上において『推敲』の開発を始めた¹⁾。しかしながら、パソコン上で文章を作成することが一般的となり、また、大型機のTSS端末では十分なユーザインタフェースが実現できないなどの理由から、パソコン NEC PC-9801 のMS-DOS上に『推敲』を実現した^{2,3)}。

2.2 概要

パソコン版『推敲』の主な機能を図1のメニューで示す。『推敲』を用いて、図1のメニューにあるような問題点について文章を検討することができる。

マクロコマンドメニュー		
A. 受身	K. カタカナ表記のゆれ	T. ユーザ-マクロコマ
B. 指示詞	L. 単語の統一	X. 基本コマンドモード
C. そのたて詞	M. 文節の統一	2. 『推敲』終了
D. 接続助詞「が」	N. 文節の特徴	
E. 注釈すべき文字列	O. 長いひらがな列	
F. 「よ」による不定表現	P. 長いひらがな列	
G. 不定表現を複数含む文	Q. 文字列毎の文字列表示	
H. 非別名文字列	R. 文章の文字列検索	
I. 見出し	S. 受身、指示詞、とりたて詞、接続助詞「が」	

図1 『推敲』のメニュー画面

入力となる文章の長さが実用的な規模(約1万字, 論文誌刷り上がり 7~8 ページ程度)であれば, 図1の各コマンドを1秒程度の時間で実行できる⁴⁾。

また、画面出力にマルチウインドウ(キャラクターベース)を採用し、色や反転表示、高速スクロールを利用するなど(図2)、ユーザインタフェース向上に努めている。『推敲』の詳細については、文献1)~7)を参照されたい。

No.	コマンド	文字列参照	行数	メモリ	処理
9	(スピード)		13	(4)	(4)
10	(ソフトウェア)		17	(8)	(24)
11	(ツール)		41	(3)	(10)
12	(データ)	シキツブ No. (1 - 27)?	51	(3)	(1)
13	(ネック)		52	(3)	(1)
14	(ハードウェア)		53	(8)	(29)
15	(パーソナルコンピュータ)		82	(11)	(3)
16	(パソコン)		85	(4)	(18)

No.	コマンド	K	W	C	参照	行数
48	: 日本語文章推敲支援ツール				における変身形の抽出	73
49	: 日本語文章推敲支援ツール				『推敲』における文字	74
50	: 日本語文章推敲支援ツール				のやりとりを行う。飯	77
51	: 飯のやりとりを行う。飯				のやりとりを行う。飯	54
52	コン上へ移植する際のネック				となったこと、逆に	32
53	: ハードウェア				の特徴を活かしつつ移	1
54	たインタフェースは: ハードウェア				に依存することによっ	9
55	たわれわれが考えた: ハードウェア				の特徴を活かしつつ移	15

図2 『推敲』の画面出力例

3. 移植の際に生じた問題

われわれは PC98 版『推敲』を、MS-DOS が動作する他のパソコン (IBM 5550, TOSHIBA J3100, FUJITSU FMR60・・・移植順) 上に移植した。PC98 版『推敲』は、システムの大部分を MS-DOS 上に構築しているが、画面出力とキーボード入力を処理する部分は、PC98 のハードウェアに依存した構造となっている。けれども、ハードウェアに依存した部分には、アプリケーションとハードウェアの間に1つの層を設けているので、その部分だけを各ハードウェア用に変更するだけで移植できる予定であった。実際に移植作業をすすめてゆくと、ハードウェアの差によるいくつかの問題が生じた。

ハードウェアの差には、まず、機能の有無といった差がある。そのような差によって生じた問題には、PC98 版『推敲』が PC98 固有の機能を利用していたために、その機能のないパソコン上へ移植する際のネックとなったこと、逆に、PC98 にはない機能を他のパソコンが持っているにもかかわらず、その機能を利用できなかったことがある。前者の例として、色の問題があった。『推敲』では色分けによって、ユーザが情報を区別できるようになっているけれども、移植対象のパソコンには、色表示のできないものがあった。後者の例は音である。『推敲』は、ユーザに確認を促す際や、入力ミスを指摘するときに音を鳴らすようになっている。PC98 では単一の音しか発生できないので、『推敲』は単に‘音を出す’という設計だった。移植を行ったパソコンには、さまざまな音を発生できるものがある。音を鳴らす理由によって音が異なるようにすれば、よりよいユー

A method for developing portable application software making use of specific hardware characteristics
 Masanori KURATA, Akira SUGANUMA, and Kazuo USHIJIMA
 KYUSHU University

ザインタフェースを提供できる。

機能の有無とは異なるハードウェアの差には、スピードの違いがある。ハードウェアのスピードの差は、ユーザインタフェースの設計にさまざまな影響をおよぼす。たとえば、画面出力が十分に速いハードウェア上で画面スクロールを実現するなら、1行単位のスクロールを用意するだけで差し支えない。しかし、画面出力が遅ければ、数行単位や画面単位のスクロールを可能にしないと、使いにくいアプリケーションソフトウェアになってしまう。『推敲』は、画面出力のスピードが十分に速いことに依存した設計であった。スピードの差は、機能の有無といった差に比べて気がつきにくいので注意しなければならない。

4. ハードウェアの特徴を活かしなが移植性のよいアプリケーションソフトウェアを開発する方法

一般に、移植性を考慮してアプリケーションソフトウェアを設計しようとすると、ハードウェアの差による問題を避けるために、対象となる複数のハードウェアの機能のうち、共通となる機能(図3のA)だけを利用することが多い。しかしその方法だと、各ハードウェアの特徴を活かすことができない。そこで、対象となる複数のハードウェアの機能をすべて含むような機能(図3のB)を持つ仮想ハードウェアに対してアプリケーションソフトウェアを設計することを提案する。

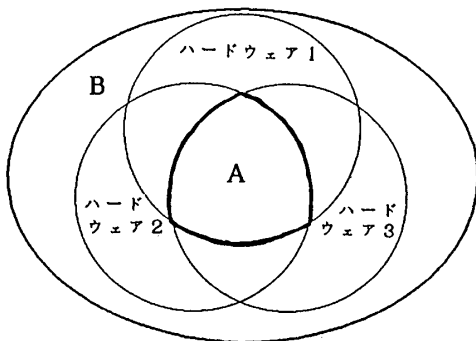


図3 仮想ハードウェアの設定

このように設計すれば、各ハードウェアの特徴を利用できるアプリケーションソフトウェアとなる。ただし、移植性を損なわないようにするため、アプリケーションソフトウェアの本質的な部分(それがなければ、アプリケーションソフトウェアが成り立たないような部分:たとえば、『推敲』の色分け)の設計は共通の機能だけを用いるようにし、各ハードウェアに共通でない機能を利用するのは、本質的でない部分(それがなくとも、アプリケーションソフトウェアを使用するのに重大な支障がない部分:たとえば、『推敲』の音)だけにする。このようにアプリケーションソフトウェアを設計するときは、その構造を図4に示すようなものにすればよいだろう。

まず、仮想ハードウェアを設定し、その上でアプリケーションソフトウェアを設計する。アプリケーションソフトウェアは、できるだけ抽象度の高い情報(たとえば、確認を促す音を出せ、とか、入力ミスを指摘する音を出せなど)で、仮想ハードウェアとデータのやりとりを行う。仮想ハードウェアと実際のハードウェアとの間のインタフェースは、解釈層と接続層に分かれている。解釈層は、個々のハードウェアの特徴に応じて、アプリケーションソフトウェアの抽象的情報を具体的情報(ピポと

いう音を出せ、ピーという音を出せ)に変換する。接続層は、解釈層と実際のハードウェアとの間で、具体的な情報をやりとりする。

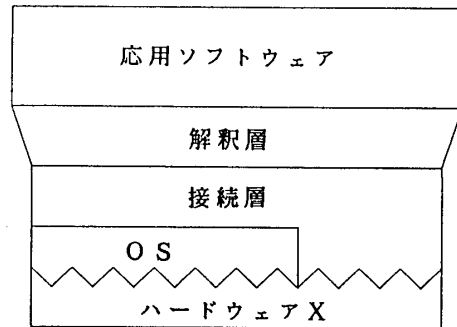


図4 アプリケーションソフトウェアの構造

このような構造にすれば、個々のハードウェアごとに、解釈層、接続層を用意することによって、アプリケーションソフトウェア本体の変更を行わずに移植が可能になる。また、接続層を一度作ってしまえば、他のアプリケーションソフトウェアを開発したときに、解釈層を用意するだけでよい。

5. おわりに

『推敲』の移植を行ったことがきっかけで、これまでに述べたようなことを考えた。現在われわれは、パソコン版『推敲』の全面改訂を計画している。改訂版『推敲』は、さまざまなパソコンのそれぞれの特徴を活かせるものにした。

謝辞

移植の際、日本アイビーエム(株)、富士通(株)に、機器の使用について便宜を図っていただいた。東芝(株)河田勉氏からは、J3100の技術マニュアルについてご教示いただいた。

参考文献

- 1) 牛島和夫他: 日本語文章推敲支援ツール『推敲』の試作とその作成環境, 情処ソフトウェア工学研究会, 1984, 35-2
- 2) 倉田昌典他: 日本語文章推敲支援ツール『推敲』のパーソナルコンピュータでの実用化, 情処第35回全国大会講演論文集, 1987, 1E-3, pp. 2459-2460
- 3) 倉田昌典他: 日本語文章推敲支援ツール『推敲』のパソコン上での実現と使用, 情処第29回プログラミングシンポジウム報告集, 1988, pp. 45-54
- 4) 倉田昌典他: 日本語文章推敲支援ツール『推敲』における応答時間, 情処第37回全国大会講演論文集, 1988, 6B-2, pp. 1008-1009
- 5) 日本語文章推敲支援ツール『推敲』(Ver1.5)使用手引書, 九州大学工学部情報工学科計算機ソフトウェア研究室, 1988
- 6) 牛島和夫他: 日本語文章推敲支援ツールにおける受身形の抽出法, 情処論文誌, 1987, Vol. 28, No. 8, pp. 894-897
- 7) 菅沼明他: 日本語文章推敲支援ツール『推敲』における字面解析手法とその評価, 情処自然言語処理研究会, 1988, 68-8