日本語プログラムの可読性の検定

3T－7

中川正樹、早川栄一、玉木裕二†
東京農工大学工学部

1．はじめに

我々は，2バイト固定長日本語文字コードの計算機システムを研究開発環境とし，その上で日本語プログラミングを実践し，その効果の評価を始めた．なお，本稿で言う日本語プログラミングとは，曖昧性を有する自然言語で処理（論理）を記述するのではなく，文字種として母国語が制限なく使用できる既存プログラミング言語によるプログラミングを意味する．この環境で相当規模のソフトウェアを開発してきた．日本語プログラミング環境は，概念設計からプログラムに至る段階的詳細化を可能にした．一方で，プログラムの可読性評価実験を行った．その結果，日本人にとっては日本語プログラムの可読性が高いことを検定した．

2．計算機システムの日本語化の背景

プログラミング言語を日本語化した例としては，COBOL，Pascal，SNOBOL，CLUなどがある[5]．しかし，一言語での日本語化には限界がある．これは，他言語プログラムとのリンケージの必要性を考えれば明らかである．例えば，ある言語で日本語識別子が使えても，リンカやソフトウェアツールが対応していなければ，日本語識別子を使うことは逆に障害になる．

OSレベルから日本語化しても，それが不完全なら一時しのぎにはなっても，かえって問題を複雑にする．商用システムでは，既存システムのコード系に日本語文字コードを混在させて日本語を表現している．しかし，そのために文字列処理が複雑になったり，日本語の使用が制限されることになる．これが，アプリケーションの国際化，地域化を遅らせる要因になっている．最近，米国製OSがマルチバイト化している理由がここにある．

いくつかのプログラミング言語で日本語化がなされてきたにもかかわらず，あまり成功をみなかった理由には，日本語以外の問題が大きくのしかかっている．日本語化の効果を純粋に議論するには，それ以外のところで重荷を負わせない環境でなければならない．

多字種文字を扱うのに，マルチバイト固定長コードを採用すれば，計算機システムの母国語化は系統的に実現できる[3～5]．母国語が使えれば，ソフトウェア開発の生産性や品質の向上が期待できる．さらに，固定長コード体系間ならば，ソフトウェアを翻訳するのはプログラミングの問題ではなくなり，ソフトウェア技術者の労力を必要としない[4]．以上の帰結として各民族が優れたソフトウェアを自分達のために円滑に相互利用できるようになるはずである．

3．日本語プログラミングの実践

現在，研究室では，システムプログラミングとアプリケーションの区別なく日本語プログミングを実践している．ここでは，オンライン手書き日本語文字認識システムの開発を例にとり，日本語プログラミングの実際的効果を考察する．

本システムの開発では，概念仕様書から，設計仕様書，関数仕様書，および部分的ではあるが，PADによる図示表現を経て，日本語プログラムへと詳細化した．OS/omicronの日本語プログラミング環境[1]は，概念設計からプログラムに至る段階的詳細化を可能にした．仕様書におけるキーワードは最終的プログラムに関数名，変数名，データ構造名などの形で反映されている．また，仕様書とプログラムの一体化は保守性にも寄与している．

この環境で作られるプログラムの特徴として，コメントの付け方が大きく変わった[2, 5]．その大半が関数やモジュールの先頭で仕様を記述するために付けられるようになった．日本語での仕様記述の段階的詳細化が関数などの仕様記述に引き継がれること，プログラミングでの補足的説明が日本語だと苦なく行えることが，仕様記述につながったと言えよう．一方，プログラムの右で処理の説明のために使われるコメントはほとんど不要になった．

4．可読性の評価実験

プログラムの可読性に関して，先の報告の後，再実験を行った．まず，実用されているプログラムからまとまりのよい部分を取りだして，例題を2つ用意した．1つは，コメント計数プログラム（約150行）で，もう一方は，表記をチェックするプログラムから抽出した．後者は，指定された前後関係で出現する数字列を検出するプログラム（約110行）である．プログラム単独の可読性を評価する目的から，それらのコメントはすべて削除した．それぞれのプログラムについて，日本語で記述したものと英語で記述したものを用意した．英語版については英語を母国語とする修士学生に手を入れてもらった．これらを，コ＿日，コ＿英，数＿日，数＿英 とする．

研究室の内外から学部4年生，大学院生を被験者とし，G1 と G2 の2つのグループに分けた．さらに G1 を G1.1 と G1.2 に，G2 を G2.1 とG2.2 に分けた．グループ分けでは，過去のプログラム歴や研究分野を参考にして，能力や知識でグループ間に偏りを生じないよう配慮した．G1.1 の被験者には，コ＿英 と 数＿日をこの順に，G1.2 には同じものを逆順に，G2.1 の被験者には，コ＿日，数＿英 をこの順に，G2.2 には同じものを逆順に与えた．被験者に

†)現在，㈱東芝システム・ソフトウェア技術研究所

A Test on the Readability of Programs in Japanese.

Masaki NAKAGAWA, Eiichi HAYAKAWA, Yuji TAMAKI, Dept. of Computer Science, Tokyo Univ. of Agriculture and Technology.

表1．プログラムの正しい読解に要した時間

グループ		被験者	研究分野	課題：数_日 読み時間(分)	回答時間(分)	合計(分)	順序	課題：コ_英 読み時間(分)	回答時間(分)	合計(分)
G1	G1.1	1	AP	7	8	15		8	8	16
		2	S	10	3	13		16	1	17
		3	S	15	*	15		40	1	41
		4	AP	11	2	13	←	20	4	24
		5	AP	7	9	16		18	12	31
		6	AP	22	*	22		30	19	49
		7	S	13	18	18		13	11	24
	G1.2	8	AP	17	3	20		19	12	31
		9	S	10	2	12	→	14	9	23
		10	AP	12	6	18		9	46	55
				課題：コ_日				課題：数_英		
G2	G2.1	11	S	11	10	21		30	5	35
		12	S	7	2	9		12	3	15
		13	AP	10	10	20		15	*	15
		14	S	2	9	31	→	29	4	33
		15	AP	16	3	19		30	1	31
		16	AP	20	*	20		23	6	29
	G2.2	17	S	10	5	15		20	20	40
		18	AP	9	2	11		16	2	18
		19	AP	35	*	35	←	44	*	44
		20	S	20	6	26		18	6	24

*: 読み時間と回答時間を分離して記録せず．
研究分野における，AP は application，S は systems software.

は，それぞれのプログラムについて，読むのに要した時間，理解を確認するための設問に答えるのに要した時間を記録するように指示した．表1に，設問に正しく答えた被験者について結果を示す．グループ間で被験者数に不均衡があるのは，正解が得られなかった被験者を除いたためである．プログラムを読む時間と，設問に答える時間を別個に見た場合，有意なことは言えない．これは，プログラムを十分理解して設問に答えるタイプと設問を見てからプログラムを見直すタイプの個人差が大きいためであろう．そこで両方の時間の合計で見ることにする．また，G1.1 と G1.2，G2.1 と G2.2 のそれぞれで，実験順序による有意差はなかったので，G1 と G2 にそれぞれ統合する．図1，図2に，以上の統合結果の読解時間分布を示す．

2つのプログラムのそれぞれについて，日本語のほうが英語より読み易いかどうか検定する目的で，統計量 t を計算し，自由度18の t 検定を行なった．コメント計数のプログラムについては，t = 2.1 となり，2.5 % の危険率で日本語のほうが早く理解できると言える．数字列検出のプログラムについては，t = 3.6 となり，0.25 % の危険率で日本語のほうが早く理解できると言える．両方の結果から，被験者によらず，日本語プログラムの方が早く理解できる，つまり，読み易いと言える．

5．おわりに

ソフトウェアの国際性は重要ではあるが，最初から国際互換性を考えて不必要な制限を課すより，まず母国語でよいものを作って，しかる後に翻訳を考えるべきである．

長期的な視点に立てば，各民族が母国語でソフトウェアを開発できることが現実的に重要であり，それらの翻訳を妨げないシステムアーキテクチャが不可欠となろう．

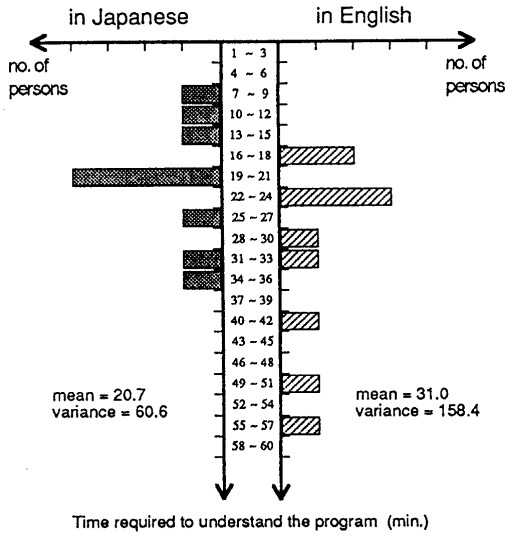

図1．コメント計数の解読時間の分布

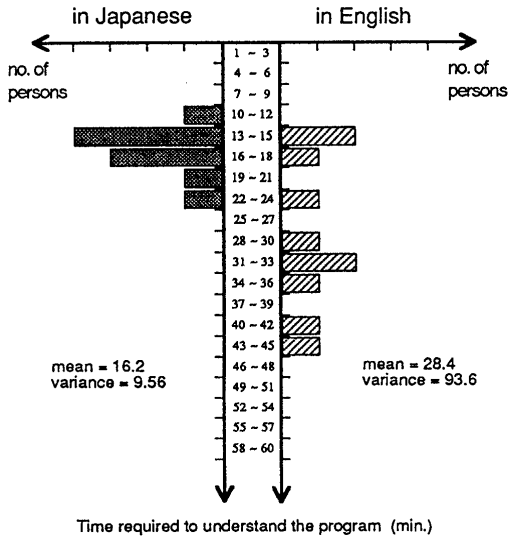

図2．文脈付き数字列検索の解読時間の分布

参考文献

[1] 鈴木他：OS/omicron における日本語プログラミング環境，情処学論，30，1，2-11 (1989).

[2] 中川他：日本語プログラミングのヒューマンファクタの一考察、信学会1991春季全大、D-72 (1991).

[3] Souya, T. et al.: Programming in a Mother Tongue: Philosophy, Implementation, Practice and Effect, Proc. 15th IEEE COMPSAC., Tokyo, 705-712 (1991).

[4] 中川他：母国語プログラミングへの方式，実践とその効果，情処学ソフトウェア工学研資，85-2 (1992).

[5] 中川他：日本語プログラムの可読性の評価と検討，情処学ヒューマンインターフェース研資，43-1 (1992).