ソフトウェア工学 40－6
（1985. 2. 7）

コンバータの効果とプログラム特性を考慮したプログラム移植工数モデルとその評価

金井敦　　高橋宗雄　　古山恒夫
（電電公社　横須賀通研）

1．はじめに

プログラム移植工数の定量的評価は、新規開発に対する移植の有効性の判断や移植工数の管理を客観的かつ的確に行なうために必要である。

プログラム移植工数を定量的に評価するためのモデルに関しては、これまで、移植工数をaL^b（a，b：定数、L：プログラム規模）の形で定量化したモデル[1]やプログラム移植時に使用されるコンバータの効果を考慮したモデル[2]が報告されている。しかし、前者は、プログラム規模が移植工数に影響を与える主要な要因であると仮定したモデルであり、また後者は、プログラム規模に加えてコンバータの効果を考慮しているが定量的には十分な解明が行なわれていない。

プログラムの移植では、ソースコンバータ（以後、コンバータと呼ぶ）が使用されることが多くソースコンバータの効果が大きいため、コンバータが移植工数に与える効果を無視するわけにはいかない。また、プログラムの移植工数は、移植プログラムの規模そのものよりもプログラム内に存在する変換項目数に依存する。そこで、プログラムの規模に加えて変換項目出現数及び変換項目に対するコンバータ機能の有無をプログラム移植工数を決定する要素としてとらえ、プログラム移植工数との関係を、具体的なプロジェクトで蓄積された移植データを用いて定量化し、プログラム移植工数評価モデルを求めた。

本報告では、まず実際の移植データの分析に基づき移植工数モデルの導出法を示し、次に、具体的な移植データを用いてモデルを導出する。さらに、本モデルを用いて移植工数を計算し実際の移植工数と比較することによりモデルの評価を行なう。最後に、本モデルを基に移植工数の特性を考察する。

2．モデルの導出法

2．1　解析データ

モデル導出の基礎となる、データの概略を示す。

（1）移植プロジェクトデータ

モデルの導出・検証には、当研究所で行なったLSI-DA（ LSI －Design Automation ）移植プロジェクトデータを用いた。本移植は、FORTRAN 記述プログラムの移植であり、解析対象となったプログラム規模は、実ステップで２２６Ｋｓである。本移植の作業手順は表１に示すように一般に行なわれている移植作業の手順と比べて特異なものではない。移植作業の中で使用されたソースコンバータ以外の支援ツールを表２に示す。コンバータとしては、FSCONV(Fortran Source CONVerter)が使用された。

本移植プロジェクトの工数データは、表１の各作業項目毎に集計されている。デバグ工数は、デバグ項目毎に工数が集計されている。移植時の変換項目は移植プログラム毎にその種別および件数が集計されている。

（2）コンバータ（ＦＳＣＯＮＶ）開発データ

FSCONVは、当研究所で開発された移植作業を支援するためのコンバータである。集計されていたデータは、総作成工数のみであり、本工数データとソースプログラムから変換項目に対応したコンバータ作成工数等の必要な情報を算出した。

2．2　モデルの基本的考え方

LSI-DAプログラム移植データを分析した結果以下の事が判明した。

（1）プログラム移植作業の工数（移植作業工数）の内で、手作業変換・デバグ工数（表１の手作業変換とデバグの部分）は、表３に示すように移植プログラム規模とはほとんど相関がなく、その他の部分（予備調査、ファイル変換、ドキュメント作成等、ただしアセンブラ部作成工数は除く）の工数は、表４に示すようにプログラム規模、プログラム本数およびTP（テストプログラム）本数と強い相関がある。このため、手作業変換・デバグ工数は、解析的手法で、その他の部分の工数は統計的手法でモデル化を行なう。

（2）移植作業工数に占める手作業変換・デバグ工数、アセンブラ部作成工数およびその他の部分の工数の割合は表５に示すようにプログラムによってばらつきが大きい。これは、コンバータの効果がプログラムにより異なるためであり、規模、TP本数等の移植工数誘因の他にコンバータの効果とプログラム特性を考慮する必要がある。

以上を考慮し、コンバータの作成工数を含めたトータルな移植工数の観点からモデル化を行なう。モデル化においては、各変換項目は他の変換項目と工数的に独立であると仮定する。この仮定は、ある変換項目を手作業で変換した場合にその変換項目を修正するということが他の変換項目の変換に影響しないという仮定であるが、実際は、ある変換項目を変換する過程で他の変換項目を変換するという場合も有り得るため独立とは言い難い。また、コンバータの作成工数の場合も、変換アルゴリズムとして、ある変換項目を変換する途中で他の変換項目を変換する場合もあり、独立であるとは言い難い。従って、本モデルは、変換項目の独立性という観点からは一次近似と見るべきである。

2．3　モデル化の詳細

前節の検討結果に基づき、総移植工数（Ｅ）を次の４つの部分工数の和で表わす。以後、工数の単位は全

表１．ＬＳＩ－ＤＡ移植における作業手順

項番	区分	作業項目	内容
1	予備調査	予備調査	作成言語・プログラム構造等の調査
2	ソースファイル変換	ﾌｧｲﾙ変換	対象機種上で扱えるファイルとする作業
3		言語変換	コンバータによる自動変換
4		手作業変換	コンバータおよびコンパイラのメッセージ指示による手作業変換
5		ｱｾﾝﾌﾞﾗ部作成	アセンブラ記述部を対象機種用に変換する作業
6	確認	ｺﾝﾊﾟｲﾙ・ﾘﾝｸ	コンパイルおよびリンク作業
7		TP走行・確認	ＴＰ走行・確認作業
8		デバグ	デバグ作業
9	ﾄﾞｷｭﾒﾝﾄ 作成	ﾄﾞｷｭﾒﾝﾄ 作成	マニュアル・保守説明書等の作成

表２．移植作業で使用されたツール

作業項目	ツール
ファイル変換	ＭＴ変換ツール
言語変換	ファイル変換ツール
全作業	ＪＣＬマクロファイル作成ツール
デバグ	COMMONブロック名一覧出力ツール プログラム構成一覧出力ツール 割り込み情報収集ツール サブルーチンインターフェースダンプツール ソースプログラムサーチツール ルーチントレースツール コール関係表作成ツール
手作業変換	実行時ルーチン代替処理ツール 拡張エラー処理ツール

て人時とする。

$$E = Em + Ea + Et + Es \quad \cdots\cdots\cdots (1)$$

Ｅｍ：手作業変換・デバグ工数

Ｅａ：アセンブラ部作成工数

Ｅｔ：コンバータ作成工数

Ｅｓ：その他の工数

（１）手作業変換・デバグ工数（Ｅｍ）

手作業変換およびデバグ工数は、コンバータおよびコンパイラを通した後、その診断メッセージにしたがい手作業変換する工数と実走行デバグにより変換する工数からなる。デバグ工数は、未発見変換項目の変換工数、変換ミスの修正工数および被変換プログラムのバグ（オリジナルバグ）の修正工数から構成される。未発見変換項目の変換は手作業で行なうことになるので手作業変換と同様に扱うことができ、Ｅｍを基本手作業変換工数（Ｅｒ）、変換ミスの修正工数（Ｅｂ）およびオリジナルバグ修正工数（Ｅｏ）の和で表わす。

$$Em = Er + Eb + Eo \quad \cdots\cdots\cdots (2)$$

Ｅｒ：基本手作業変換工数（確認のためのＴＰ走行・コンパイル・リンクを含む）

Ｅｂ：変換ミスの修正工数（確認のためのＴＰ走行・コンパイル・リンクを含む）

Ｅｏ：オリジナルバグ修正工数

（ａ）基本手作業変換工数（Ｅｒ）

基本手作業変換工数（Ｅｒ）は、手作業変換項目を変換するのに要する工数である。Ｅｒを変換作業工程毎に分割し①バグ原因究明工数、②変換方法検討工数③変換工数および④確認工数に細分して、その工数の和とした。Ｅｒはコンバータを使用しないで手作業変換した場合の変換項目一件当たりの工数（完全手作業変換工数（Ｅｒｉｊ））を中心に、変換項目出現数（ｎｒｊ）が増加するとそれに比例して増加し、コンバータ機能の効果（コンバータ機能の効果（Ｃｃｉｊ））があればそれにより減少する。ここで、ｉ，ｊはそれぞれ変換過程の番号と変換項目番号を表わしている。Ｅｒを次のように仮定する。

$$Er = \sum_{j=1}^{N} Er1j \cdot Cc1j \cdot \bar{n}rj + \sum_{j=1}^{N} Er2j \cdot Cc2j \cdot \bar{n}rj + \sum_{j=1}^{N} Er3j \cdot Cc3j \cdot nrj + \sum_{j=1}^{N} Er4j \cdot Cc4j \cdot \bar{n}rj \quad \cdots\cdots (3)$$

表３．　手作業変換・デバグ工数の分析結果
（説明変数：プログラム規模）

	平方和	自由度	不偏分散	Ｆ値	検定結果
回帰	4.45*10	1	4.45*10	1.16	
誤差	1.19*10	5	3.82*10		
合計	2.36*10	6			

表４．移植作業工数－手作業変換・デバグ工数
－アセンブラ部作成工数の分析結果
（説明変数：プログラム規模、プログラム本数、ＴＰ本数）

	平方和	自由度	不偏分散	Ｆ値	検定結果	純平方和	寄与率(%)
回帰	7.01*10	3	2.34*10	968.9	**	7.00*10	99.7
誤差	7.24*10	3	2.41*10			1.45*10	0.3
合計	7.02*10	6				7.02*10	100

表５．部分工数の占める割合

サンプル	Ｅｍ－Ｅｖ（%）（手作業変換・デバグ）	Ｅａ（%）（ｱｾﾝﾌﾞﾗ部作成）	Ｅｓ'（%）（その他）
A	25.8	0	74.2
B	7.5	0	92.5
C	21.3	28.9	49.8
D	56.6	0	43.4
E	37.8	0	62.4
F	12.1	7.9	80.0

N　：変換項目数

$\bar{n}rj$：nrjが0でないとき1、0のとき0の値を取る。

nrj：変換項目の出現数

Erij ：完全手作業変換工数

コンバータを使用しない場合の手作業変換工数である。

変換項目番号	1	2	・・・	j	・・・	N
バグ原因究明工数	Er11	Er12	・・・	Er1j	・・・	Er1N
変換方法検討工数	Er21	Er22	・・・	Er2j	・・・	Er2N
変換工数	Er31	Er32	・・・	Er3j	・・・	Er3N
確認工数	Er41	Er42	・・・	Er4j	・・・	Er4N

Ccij ：コンバータ機能の効果

コンバータを使用したことにより工数が軽減される割合（コンバータを使用した変換工数／完全手作業変換工数）である。

変換項目番号	1	2	・・・	j	・・・	N
バグ原因究明工数	Cc11	Cc12	・・・	Cc1j	・・・	Cc1N
変換方法検討工数	Cc21	Cc22	・・・	Cc2j	・・・	Cc2N
変換工数	Cc31	Cc32	・・・	Cc3j	・・・	Cc3N
確認工数	Cc41	Cc42	・・・	Cc4j	・・・	Cc4N

$0 \leqq Ccij \leqq 1$

Ccij は作業の必要のある場合は1、必要のない場合は0、修正要求メッセージ等により作業量が軽減されるような場合は1から0の間の中間値を取る。移植環境に応じて、表6のように各要素の値を設定する。

（b）変換ミスの修正工数（Eb）

手作業変換した結果、修正ミスを犯しそのミスを修正する工数である。

$$Eb = Aeb \cdot Beb \cdot Nrs \quad \cdots\cdots\cdots (4)$$

Aeb：修正ミス発生確率

Beb：一件当たりのバグ原因究明・変換・確認工数

Nrs：手作業修正件数

（c）オリジナルバグ修正工数（Eo）

オリジナルバグは存在しないものとする。

（2）アセンブラ部作成工数（Ea）

アセンブラ部作成工数は、被変換プログラムのアセンブラ規模（La）に比例するものとした。

$$Ea = Aea \cdot La \quad \cdots\cdots\cdots (5)$$

La：アセンブラプログラム規模

Aea：定数

（3）コンバータ作成工数（Et）

コンバータの構成は、図1に示すように、大部分のものは、変換項目に対して共通な部分（Ecom）と変換項目に依存する部分（Eyj）から成るものと考える。

$$Et = \sum_{j=1}^{Na} Eyj + Ecom \quad \cdots\cdots (6)$$

$$= \sum_{j=1}^{Na} (Eyj + Ecom / Na) \quad \cdots\cdots (7)$$

Eyi　：j番目の変換項目に依存する部分の工数

Ecom ：変換項目に対して共通な部分の工数

Na　：コンバータの対象変換項目数

（4）その他の工数（Es）

予備調査、ファイル変換、言語変換、コンパイル、リンク（JCL作成、1回走行）、TP走行・確認（JCL作成、1回走行）およびドキュメント作成の工数であり、プログラム規模（L）、プログラム本数（Np）、TP本数（Ntp）と強い相関があるため、これらの一次結合として考える。

$$Es = Aes \cdot L + Bes \cdot Np + Ces \cdot Ntp + Des \quad \cdots\cdots\cdots\cdots (8)$$

L　：プログラム規模

Np ：プログラム本数

Ntp：TP本数

Aes，Bes，Ces，Des：定数

以上のモデルを構成している各要素をトリーで表現したものを図2に示す。同図に示すように、各構成要素は、①移植プログラムにより決定されるもの、②コンバータ、言語により決定されるもの、および③要員の技術レベル、支援ツール（コンバータは除く）に影響されるもの（ここでは定数）にクラス分けされる。一般に、移植環境が決定しているというのは、②③が決定しているという意味である。

3．　適用例

3．1　具体的モデルの導出

本節では、2．1節で示した移植データを用いて、2．3節に従いLSI-DA移植における移植工数モデルを決定する。この例は、FORTRAN からFORTRAN への移植であり、コンバータとしてFSCONVを使用してる。支援ツール、移植手順は表1、2に示すような環境の場合のモデルとなっている。

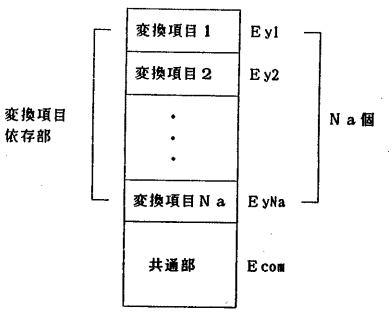

図1．　コンバータの構成

（１）手作業変換・デバッグ工数（Ｅｍ）

（ａ）基本手作業変換工数（Ｅｒ）

ＬＳＩ－ＤＡの移植データからは、変換項目に対応した工数は、デバグ時の工数を除いて得られなかった。そこで、Ｅｒｉｊ はある程度勘に頼って決定しなければならなかった。各工数は、デバグ時発見項目（１１項目）を合わせて、９９項目となった。Ｅｒｉｊ の例を表７に示す。Ｃｃｉｊ は、コンバータの機能を基に表６から決定した。表８にＣｃｉｊ の例を示す。

（ｂ）変換ミスの修正工数（Ｅｂ）

変換ミス発生確率は、移植データから、１０件当たり１件とした。修正工数は、工数データを基に、バグ原因究明工数４、修正工数０．５、確認工数３として計算した。

Ｅｂ＝0.1 ・7.5 ・Ｎｒｓ

＝0.75・Ｎｒｓ　　・・・・・・・・・・(9)

（２）アセンブラ部作成工数（Ｅｓ）

工数データより、定数を次のように設定した。

Ｅａ＝350 ・Ｌａ　　（Ｌａ：Ｋｓ）・・・・(10)

（３）コンバータ作成工数（Ｅｔ）

各変換項目に寄与しているソースプログラムの記述量をカウントし、全体の規模との比からＥｙｊとＥｃｏｍを求めた。変換項目と重複している部分は、重複している項目数で割りそれぞれの変換項目の寄与分とした。表９に、ＥｙｊとＥｃｏｍ の例を示す。Ｎａは、８８となった。

（４）その他の工数（Ｅｓ）

基本検討で行なった重回帰分析結果を用いた。この統計解析に用いたデータは、このモデルにおいてＥｍに含まれているものとした修正結果確認のためのコンパイル・リンク・ＴＰ走行・確認工数（Ｅｖ）を含んでいる。この部分の工数はデータからは分離不能であった。ここでは、便宜上ＥｖをＥｓに含めたＥｓ’を定義する。

Ｅｓ’＝Ｅｓ＋Ｅｖ　・・・・・・・・・・・(11)

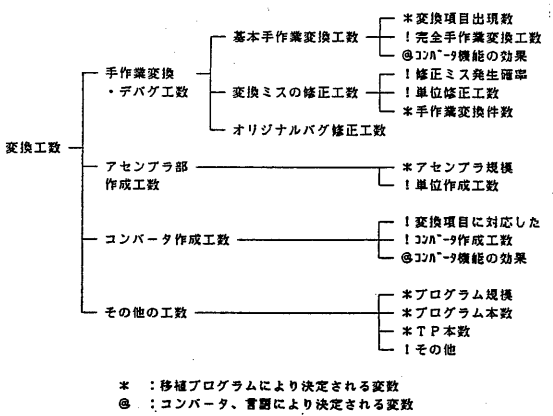

図２．　移植工数の構成要素

表６　　コンバータ機能の効果ＣＣｉｊ の決定法

項番	状態		作業	バグ原因究明	変換方法検討	変換	確認	備考
1	コンバターの対象外	コンバイラ検出可	変換方法未知	0	1	1	0．5	
2			変換方法既知	0	0	1	0．5	すでに体験した変換
3		コンバイラ検出不可	変換方法未知	1	1	1	1	
4			変換方法既知	1	0	1	1	
5	コンバータの対象	自動変換可	確認不要	0	0	0	0	
6			確認要	0	0	0	1	確認要メッセージ
7		自動変換不可	変換方法既知	0	0．5	1	1	
8			変換方法未知	0	1	1	1	サブルーチン作成等

表７　完全手作業変換工数Ｅｒｉｊ の例

項番	変換項目内容		工数			
	変換対象（変換前）	変換後	バグ原因究明 Ｅｒｌｊ	変換方法検討 Ｅｒ２ｊ	変換 Ｅｒ３ｊ	確認 Ｅｒ４ｊ
1	INTEGER*2 の要素に対するストリング定数での初期設定（ I2/'ABC'/ ）	I2/2'4142'/	コエメンラッパイセイーラジ	1	0．5	3
2	計算型GOTO文番号並びの個数が101 個以上（GO TO(K,K,**))	GOTO (K,K,**),i GOTO(K,K,**),i-100	コエメンラッパイセイーラジ	1	1	3
3	内部コードの計算をEBCDICコードで計算している	JIS コードで計算する	１０	3	0．5	3
4	ENCODE,DECODE 文が使用されている	サブルーチンにする	コエメンラッパイセイーラジ	１０	0．5	3

表８コンバータ機能の効果Ｃｃｉｊ の例

変換項目 表７の項番	バグ原因究明Ｃｃｉｊ	変換方法 Ｃｃ２ｊ	変換 Ｃｃ３ｊ	確認 Ｃｃ４ｊ
1	0	0	0	0
2	0	0	0	0
3	1	1	1	1
4	0	0．5	1	1

解析結果を以下に示す。

$$Es' = 3.57L + 65.1Np + 3.76Ntp + 27.8 \quad (L : Ks) \cdots\cdots (12)$$

以上の結果から、移植工数モデルは、次式で表わされる。

$$E = Em + Ea + Et + Es = (Em - Ev) + Ea + Et + Es'$$

$$= \sum_{j=1}^{99} Er1j \cdot Cc1j \cdot \bar{n}rj + \sum_{j=1}^{99} Er2j \cdot Cc2j \cdot \bar{n}rj + \sum_{j=1}^{99} Er3j \cdot Cc3j \cdot nrj + 0.75Nrs + 350.0\ La + \sum_{j=1}^{88} (Eyj + 40.8)$$

$$+ 3.57L + 65.1Np + 3.76Ntp + 27.8 \quad (人時) \cdots\cdots (13)$$

上式に於いて、定数Erij，Ccij，Eyjは表7、表8、表9に例を示すように、各変換項目毎に値が定められてる。nrj，Nrs，La，L，Np，Ntpは、被変換プログラムにより決定される変数である。（Em－Ev）は、コンバータによる確認工数削減の割合（Cc4j）を0に設定し確認工数を除外することにより求めている。

3．2　モデルの評価

前節でもとめたモデルを用いて、LSI-DAプログラムの移植作業工数（移植工数からコンバータ作成工数を引いたもの）を計算し、実際の移植工数と比較する。計算に際して、移植プログラムの変換項目が既知のものとした。また、ここでは、コンバータ作成工数は除外して計算した。入力として与えたデータを次に示す。

①プログラム規模（L）
②プログラム本数（Np）
③TP本数（Ntp）
④変換項目出現数（nri）
⑤アセンブラ規模（La）

プログラム規模を横軸にした、計算値と実測値の比較を図3に示す。同図の直線は、移植作業工数が規模に比例するものとして、最小二乗法で求めたものである。計算値と実測値の差は、本モデルでは平均12％であるのに対し、最小二乗法では48％である。本モデルによる評価がより正確であることがわかる。

計算値と実測値の誤差の原因は、主に、Erij の設定値が不適当なためと思われる。

4．　移植工数特性の考察

本モデルを用い、応用例として、①コンバータを援用した場合の移植工数削減効果、②コンバータを援用して採算のとれる最小のプログラム規模を推定した結果について先に報告した[4]。本章では、本モデルを用いて移植工数の特性の考察を行なう。

移植工数を規模の関数として表わしたモデルが参考文献（1）に示されてる。同文献によると、図5に示すように移植工数は、aL^b（$b<1$）という形で示され、移植プログラム規模が大きくなるほど単位規模当たりの移植工数は小さくなっている（移植効率は高くなっている）。この傾向を、本モデルで説明すると、移植プログラム規模が大きくなるほど、性能の良いコンバータが使用されるためであると考えられる。プログラム移植において最適なコンバータを使用したと仮定した場合のプログラム規模と移植工数の定性的な関

表9　コンバータ作成工数例

項番	Eyi（人時）	Ecom（人時）	和（人時）
1	61.1	40.8	101.9
2	44.1	40.8	84.9
3	0	0	0
4	17.1	40.8	57.9

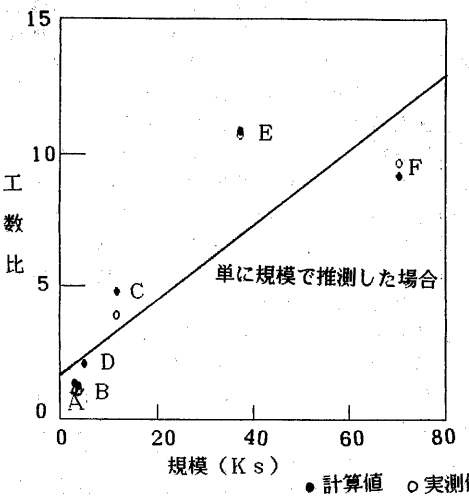

図3．　移植工数の計算値と実測値の比較

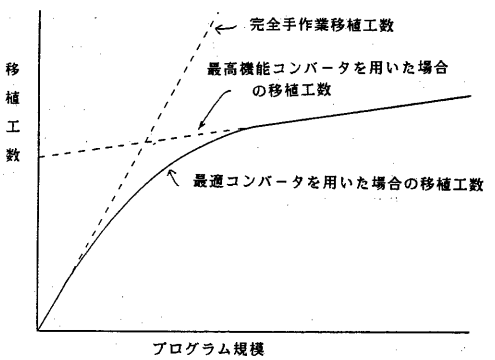

図4．　最適コンバータ使用時の移植工数

係は図４に示すような曲線となると考えられる。すなわち、移植プログラム規模が小さいときには、完全手作業変換をした場合の工数に従い、大規模プログラムの場合は、最高機能のコンバータを使用した場合の工数に従う。その中間的な規模のプログラムを移植する場合は、そのプログラムの規模や性質に応じて最適なコンバータを使用することになる。なお、最適なコンバータというのは、本モデルを用いて各変換項目毎に手作業で変換した場合の移植工数とコンバータを用いたときの工数（コンバータ作成工数も含む）を比較し、少ないほうを選択した場合の、コンバータ機能を意味している。ここで、注意を要することは、移植工数は規模だけで議論できず、プログラムの特性（変換項目出現率）を考慮しなければならない点である。例えば、移植性を考慮して、互換のある表現で記述した場合は、規模に係わりなくソース変換工数は０となる。従って、このためこのような移植工数の考察は、移植プログラムの特性が一定値の場合のものであり、全体的な平均の傾向を表わしているとみるべきである。個々の移植プログラムの移植工数はこのような全体的傾向から大きくずれる場合がある。

２章で示したモデルを用いて最適コンバータを使用する場合の移植工数を求めたものが図５である。この計算を行なうに当たり次のような仮定をしている。プログラムの特性は一定である（これまで解析したサンプルの平均の特性とした）。また、モジュール数はプログラム規模１０Ｋｓ毎に１個で、テストプログラム本数は、プログラム規模２Ｋｓ毎に１本で計算した。

実際に計算した曲線は、図５に示すように、図４のような傾向は顕著には表われていない。これは、①サンプルとしたプログラムの変換項目出現数分布が片寄っていたこと、②ソースプログラム変換が占める工数が予想以上に小さいことが考えられる。Wolberg のモデルが正しいとすれば、ソース変換以外の工数における工数削減効果があるものと思われ、削減効果の原因は、支援ツールの使用、要員の熟練レベルの向上等がさらに考えられる。

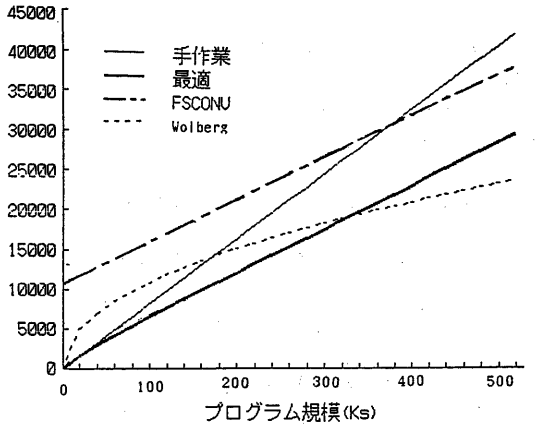

図５．　移植工数の計算例

５．おわりに

コンバータの効果とプログラム規模を考慮した移植工数モデルの導出法と、具体的なモデル導出例およびその評価を示した。続いて、本モデルを用いた移植工数モデルの特性の検討を行なった。

本モデルは、プログラムの特性（変換項目数分布）を考慮しているため個々のプログラム毎に正確な移植工数把握ができる。また、コンバータ機能の効果を考慮しているためコンバータが異なる種々の環境に適用できる。しかし、本モデルは以下の点で不完全なところがあり、さらに検討する必要がある。

（１）各変換項目毎に、変換工数、コンバータ作成工数が独立であると仮定している。しかし、実際は変換項目相互に絡みあっており、互いに切り放せない。その点で、本モデルは一次近似となっている。

（２）支援ツールとしては、コンバータしか考慮していない。しかし、ソース変換以外の工数が占める割合がかなり多いので、その他のツールの効果も考慮するべきである。

（３）要員の技術レベルは、データが少なくモデルに考慮することができなかった。

（４）本モデルを移植工数の評価に用いる場合、移植プログラムの変換項目をあらかじめ知る必要がある。このため、プログラムの部分サンプリング等により変換項目出現数を予測できる方法について、検討する必要がある。

（５）移植データが不足のためモデルの検証が十分にできていない。さらに、検証を進めて行く必要がある。

（６）本モデルの適用領域は同一高級言語間の移植である。今後は、他言語間あるいはアセンブラレベルの移植にも適用できるように検討して行きたい。

謝辞

本検討に当たり、有益な御意見を頂いた当研究所の川原洋人及び馬場康彦両氏に感謝いたします。

＜参考文献＞

（１）John R.Wolberg,"Comparing the Cost of Software Conversion to the Cost of Reprograming," SIGPLAN Notices,16(4),104-110(1981).

（２）John R. Wolberg,"A Costing Model for Software Conversion," Software-prac. and Exp., vol.12,1043-1049(1982).

（３）金井、高橋、古山、”プログラム移植コスト予測の一方法、”昭和５９年信学会全国大会、１７２３、ｐ．７－４５．

（４）金井、高橋、古山、”プログラム移植工数モデルの応用例、”情報処理学会第２９回全国大会（昭和５９年後期）ｐｐ．４６３－４６４．