計測自動制御学会産業論文集
Vol.6, No.5, 31/40（2007）

システム提案書記載のキーワードを用いた小規模事務処理システム開発工数見積りの一手法

高 橋 正 和*・齋 藤 達 也**
福 江 義 則***・津 田 和 彦**

A Method for the Estimation of Man-Hours in the Small Scale Office Information System Using Keywords Described in the System Proposal

Masakazu Takahashi*, Tatsuya Saito**,
Yoshinori Fukue*** and Kazuhiko Tsuda**

In the development of the Small Scale Office Information system (SOIS), the man-hours estimation is carried out in the phase of developing system proposal, and it has a large margin of error. In this situation, this paper proposes a man-hour estimation method of SOIS development using described information in the system proposal. When we proposed this method, we analyzed the relationship between the system proposals and the records of the man-hours on several dozen SOIS developments. And we derived following relationships: (a) the keywords that are described in system proposal correspond to specific input and output data of SOIS, and (b) the multiplication product of the number of the input and output data is in proportion of the man-hours for the developments. Using those relationships, the man-hours will be estimated as followings: (1) extract all keywords that is related to the man-hours from system proposals, (2) list up all input and output data corresponding to keywords, (3) calculate the multiplication products of both the number of the input and out data, (4) conduct regression analysis between the multiplication products and the records of the SOIS developments, and derive the relational expression, and (5) estimate the man-hours using the relational expression. As the result of applying this method to some developments, we could estimate the man-hours within 10[%] errors.

Key Words: system development, man-hours estimation, business system, keyword, regression analysis

1. はじめに

本論文では，Windows 系システム開発ツールを用いて開発された小規模な事務処理システムの開発工数を，システム提案書に記述された内容をもとに，見積る手法について提案する．IT 業界では，多数のシステム開発企業が設立されているが，それらの大部分は，小規模な企業である（以下，小規模開発企業と呼ぶ）．小規模開発企業が受注し，開発するシステムは多種多様であるが，代表的なものとして，小規模な事務処理システムがある．小規模な事務処理システムとは，発注企業の部門内における物品の要求，発注，検収，管理などの特定の事務処理を取り扱ったシステムである．これは，企業全体の事務処理を支援する基幹システムとは異なり，部門内の特定の事務処理を支援するシステムである．小規模な事務処理システムは，部門内の限定された範囲で使用され，他システムとの連携も少なく，単純で，規模の小さいシステムである．したがって，小規模な事務処理システムの大部分は，容易かつ安価に開発するために，Windows パーソナル・コンピュータ（以下，PC）上で，Visual Basic や Visual C++ などの Windows 系システム開発ツールを用いて開発されることが多い．以降，本論文では，このような小規模な事務処理システムのことを SOIS（Small Scale Office Information System）と呼ぶ．小規模開発企業が，SOIS を受注する際には，SOIS の機能，性能，受注金額，開発期間などをシステム提案書としてまとめ，発注企業に提示する．その内容について発注企業から合意が得られた場合には，正式な受注となり，開発が開始される．しかし，実際に SOIS 開発が始まる

* 島根大学総合理工学部数理・情報システム学科
(株) ギャラクシーエクスプレス
** 筑波大学大学院経営システム科学専攻
*** 徳島大学大学院工学研究科情報システム工学専攻
* Department of Mathematics and Computer Science, Interdisciplinary Faculty of Science and Engineering, Shimane University
Galaxy Express Corporation
** Graduate School of Systems Management, University of Tsukuba
*** Graduate School of Advanced Technology and Science, Tokushima University
（Received November 9, 2006）

と，開発費用が受注金額を大幅に上回るケースが頻繁に発生している．主要な原因の一つとして，システム提案書を作成した際の SOIS の開発に要する総時間（以下，開発工数）の見積もりに誤差が大きく，それを基に算出した受注金額の見積りの誤差が大きくなったことが挙げられる．システム開発工数の見積り手法については，2.2.2 節に述べるように様々な研究がなされているが，それらは大型計算機上で稼動する大規模なシステムの開発工数見積りを対象としたものが大半である．したがって，異なる条件のもとで開発される SOIS の開発工数の見積もりに，そのまま適用することは，困難である．そのため，小規模開発企業では，SOIS の開発工数をシステム提案書の作成時点で，正確に見積る手法が必要不可欠となっている．そこで，本論文では以下の手法を提案する．

[**開発工数見積り準備段階**　]

(1)　入力データの個数と出力データの個数の積が SOIS の処理の個数に比例し，さらに，その個数が開発工数に比例するという仮説のもと，SOIS の開発実績データ（”入力データの個数と出力データの個数の積”と”開発工数”）を回帰分析して，それらの間の関係式を求める．

(2)　既存 SOIS のシステム提案書の内容を分析して，SOIS の入出力データと関連の深いキーワードを抽出して辞書に登録する．

[**開発工数見積り段階**　]

(1)　見積り対象となる SOIS のシステム提案書から，辞書に登録したキーワードを抽出し，入力データの個数と出力データの個数を求める．

(2)　入力データの個数と出力データの個数を開発工数見積りの関係式に代入し，開発工数を見積る．

最後に本論文の構成について述べる．第 2 章では，既存の開発工数見積り手法とその問題点を記述する．第 3 章では，提案する SOIS の開発工数見積り手法の概要を記述する．第 4 章では，新規開発の SOIS に対する提案手法の適用と評価を記述する．そして，第 5 章でまとめを記述する．

2.　現状における SOIS の開発工数見積りの問題点

本章では，はじめに小規模開発企業における SOIS 開発工数の見積り手順とその問題点を記述する．つぎに既存の開発工数見積り手法と先行研究について概要を記述し，SOIS の開発工数の見積りに適用した場合の問題点を記述する．

2.1　SOIS 開発工数見積りの手順とその問題点

Fig.1 に小規模開発企業における現状の SOIS 開発工数の見積り手順示す．

はじめに，発注企業は，小規模開発企業に対して SOIS の見積り依頼を行なう．小規模発注企業は，発注企業からの提案依頼の内容と今までの開発経験をもとに SOIS のシステム提案書の一次案を作成する．**Fig.2** に示すように，システム提案書の一次案には，SOIS に対する機能，開発期間，開発費用などが記述される [4]．多くの場合，SOIS の機能は，SOIS

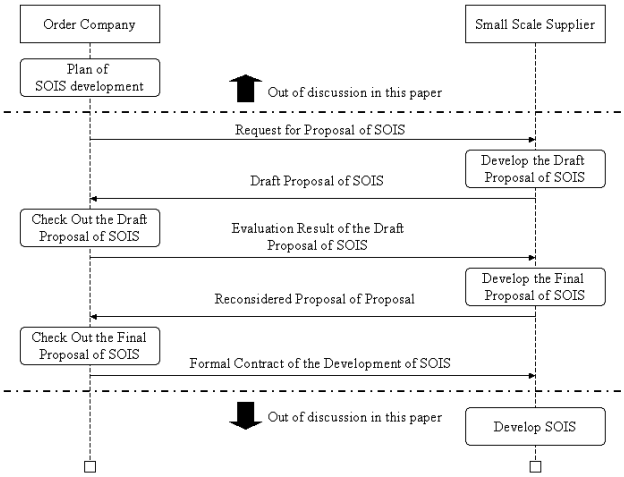

Fig. 1　Estimation process of SOIS with small scall supplier

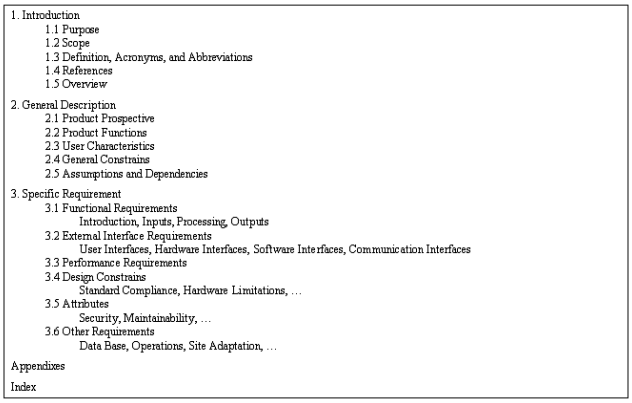

Fig. 2　Sample contents in the system proposal

開発者が，今までに開発した SOIS のなかから，発注企業の要望に最も類似しているものを選定し，それに修正を加えることで決定される．開発期間と開発工数についても，機能と同様の手法で見積られる．そして，見積られた開発工数とその単価を掛け合わせて開発原価を算出し，それに小規模開発企業の管理費や利益などを上乗せして提示価格を決定する．言い換えれば，SOIS の機能，開発期間，開発費用は，SOIS 開発者の経験に依存して，見積られているのが実情である．

二番目に，小規模開発企業は，発注企業に対して，作成したシステム提案書の一次案の内容を説明する．発注企業は，説明された SOIS の機能十分性や開発期間と提示価格の妥当性などについて評価を行い，その結果を小規模開発企業にフィードバックする．その際，必要があれば審査会を実施して，記述内容に関する質疑応答を行なう．

三番目に，小規模開発企業は，発注企業からの SOIS の評価結果を基に，機能，開発期間の再検討を行い，その結果を基に開発工数を見積り，開発原価と提示価格を算出する．小規模開発企業は，それらの結果を反映させたシステム提案書の正式版（以下，単にシステム提案書と呼ぶ）を発注企業へ提示する．この開発見積り作業はシステム提案書の一次案を

作成した場合と同様，SOIS 開発者の経験をもとに行われる．

四番目に，発注企業は提出されたシステム提案書の機能十分性や開発期間と提示価格の妥当性を確認し，それらが妥当であれば，正式な開発契約を結び，開発段階に移行する．したがって，これが，小規模開発企業と発注企業との間での正式な契約内容となる．そのため，小規模開発企業の利益を確保するためには，システム提案書に記述される提示価格を適切に算出することが必須である．

しかし，実際の SOIS 開発では，開発段階に移行した後に予期せぬトラブルが発生して開発工数が増加し，その結果，開発期間の長期化や開発原価の超過が発生する．この原因は，SOIS 開発者の経験をもとにした SOIS の機能と開発工数の見積りでは，「システム提案書に記述されたすべての機能を漏れなく抽出することが困難であること」，「根拠に基づいた精度の高い各機能毎の開発工数見積りが困難であること」などの理由によると考えられる．そのため，システム提案書の作成段階で，精度の良い SOIS 開発工数の見積り手法が必要とされている．

2.2 既存手法と先行研究の概要と適用上の問題点

本節では，開発工数見積りに関する既存手法と先行研究について記述し，SOIS の開発工数の見積りへ適用する上での問題点を記述する．

2.2.1 既存の開発工数見積り手法

本項では，既存のシステム開発工数見積り手法と，それらを SOIS の工数見積りに適用する場合の問題点を記述する．

類推法[8]は，開発済みシステムの機能・開発工数の実績をリスト化しておき，そのなかから開発するシステムの機能に最も近いと考えられるものを探し出し，実績を参考に開発工数を見積る方法である．しかし，「リスト中に開発するシステムに類似したものが存在するとは限らない」，「リストだけでは類似システムと判断できない」などの問題がある．したがって，SOIS のシステム提案書の作成段階で，類推法を用いて十分な精度の開発工数見積りを行なうことは困難である．

積算法[6),11)]は，システムを小さなサブ・システムに分解し，サブ・システム単位で必要な人月や開発費用を見積り，それらの和を計算することでシステム全体の開発工数を見積る方法である．しかし，設計情報が不足している場合，サブ・システムへの分解は困難である．したがって，SOIS のシステム提案書の作成段階で，積算法を用いて十分な精度の開発工数見積りを行なうことは困難である．

LOC（Line of Code）法[9]は，開発するシステムの LOC を見積り，それを LOC －開発工数の換算式に代入して開発工数を見積る手法である．しかし，要求仕様作成段階で LOC を見積ることが困難である上，SOIS の開発言語の主流になっている Windows 系システム開発ツールでは，ソフトウェア部品（コンポーネント）と LOC との関係が明確ではない．したがって，SOIS のシステム提案書の作成段階で，LOC 法を用いて十分な精度の開発工数見積りを行なうことは困難である．

ファンクション・ポイント法[2),8)]は，システムの有するファンクションの種類・個数・ポイント・単位ポイント当たりの開発工数を求め，その積を計算することで開発工数を見積る手法である．ファンクション・ポイントの計算ルールは International Function Point Users Group（IFPUG）により規定されている[10]．しかし，SOIS のシステム提案書の作成段階で，ファンクションを漏れなく抽出することは困難である．したがって，ファンクション・ポイント法を用いて十分な精度の開発工数見積りを行なうことは困難である．

Constructive Cost Model（COCOMO）[5]は，開発するソフトウェアの規模を KLOC で表し，モデル式を用いて，開発工数と開発期間を算出する手法である．また，COCOMOII[19]は，工数見積りのモデルを開発の段階に応じてアプリケーション組み立てモデル，初期設計モデル，ポストアーキテクチャモデルの 3 種類に分け，それぞれ個別のモデル式を用いて開発工数と開発期間を算出する．しかし，いずれの手法もモデル式の入力データとしてオブジェクト・ポイント，KLOC，ファンクション・ポイントなどが必要であり，システム提案書の作成段階で十分な精度の開発工数見積りを行なうことは困難である．

以上の結果，既存の開発工数見積り手法では，SOIS のシステム提案書の作成段階で，開発工数を精度良く見積ることが困難であることが判った．

2.2.2 開発工数見積りに関する先行研究

本項では，先行研究と，それらを SOIS の工数見積りに適用する場合の問題点について記述する．

Dolado[12]は，COBOL と Informix-4GL で記述された 10^5[LOC] 以上のソフトウェアの規模を，そこで使用されるソフトウェア部品の規模を積み上げていくことで見積る手法を提案した．しかし，この手法は大規模なソフトウェアの規模を見積る手法であり，開発言語の特徴も異なっているため，そのまま SOIS の開発工数見積りへ適用することは困難である．

Matson[13]らは，ファンクション・ポイントとソフトウェア規模との関係式を，回帰分析を用いて導出した．しかし，システム提案書の作成段階で，ファンクション・ポイントを正確に算出することは困難であるため，この手法をそのまま SOIS の開発工数見積りへ適用することは困難である．

Costagliola[14]らは，ファンクション・ポイントに類似したクラス・ポイントを用いたオブジェクト指向ソフトウェアの規模見積り手法を提案した．しかし，システム提案書の作成段階で，クラス・ポイントを正確に算出することができないため，この手法をそのまま SOIS の開発工数見積りに適用することは困難である．

Auer[15]らは，ソフトウェアの類似性に着目した開発費用見積り手法を提案した．この手法では，類似性に加えて，ソフトウェア開発形態に応じて，開発費用見積り算出式の係数を変化させることで，正確な見積りを実現する．しかし，ある程度，ソフトウェア設計が進まなければ，類似性や開発形態

を明確にすることができないため，この手法をそのまま SOIS の開発工数見積りに適用することは困難である．

Jørgensen[16]は，コスト見積りガイドラインを提案した．ガイドラインには「直感ベースの評価を信用しない」,「安易に不確かな評価モデルを適用せず，専門家の判断を尊重する」,「形式化された明確な見積りプロセスを適用する」,「複数の開発者が見積もりを行い，おのおのの結果を考慮して最終見積りを算出する」などのルールから構成される．しかし，この手法は見積りに要する作業項目が多いため，SOIS の様な小規模なソフトウェア開発に適用することは困難である．

以上の結果，先行研究をそのまま，SOIS のシステム提案書の作成段階での，開発工数見積りに適用することは困難であると考えられる．

3. キーワードを用いた SOIS 開発工数見積り手法の提案

本章では，システム提案書の作成段階で SOIS の開発工数を見積る手法について記述する．はじめに 3.1 節では，SOIS の特徴について説明する．つぎに 3.2 節では SOIS の特徴をもとにした開発工数見積り手法の仮説について記述する．さらに 3.3 節では，SOIS の開発工数見積りに使用するキーワード辞書について記述する．最後に 3.4 節では，SOIS の開発工数の実績を統計処理することで仮説の検証を行なう．

3.1 SOIS の特徴

本節では，はじめに，開発工数見積りの対象である SOIS の概要について記述し，つぎに Windows 系システム開発ツールを用いて開発された SOIS の特徴を記述する．

3.1.1 SOIS の概要

本論文で扱う SOIS とは，部門内の小規模な事務手続きをシステム化したものである．SOIS の例として購買処理や支払処理などが挙げられる．これらのシステムでは，部門内にある購入品データや支払いデータと呼ばれる入力データ（小規模なデータ・ベースを含む）から，個々の処理に必要な情報を取り出し，それらを加工して購入品一覧や支払一覧を作成し，購入品データや支払データと呼ばれる出力データ（小規模なデータ・ベースを含む）に保存する．したがって，これらの SOIS の処理は，**Fig.3** に示すように (a) データ入力処理，(b) データ加工処理，(c) データ出力処理の繰り返しとなる．

ここで，(a) データ入力処理とは，入力用データやデータ端末から対象となるデータを入力することであり，たとえば購入品データから購入品の一覧とその個数を取り出す処理が挙げられる．(b) データ加工処理とは，(a) で入力したデータに対して整理，検索，四則演算などの処理を実施することである．データ整理は，入力データを降順，昇順などの意味のある順番に並べ替える処理であり，たとえば，購入品データを購入帳票番号の若い順に並べ替える処理が挙げられる．データ検索は，整理が終了したデータのなかから，入力した条件に合致したデータを抽出する処理であり，たとえば，全購入品データのなかから特定の購入先に発注するデータを抽出する処理が考えられる．データ四則演算は，検索したデータに対して目的に応じた計算を行なう処理であり，たとえば購入先毎に購入品の金額を計算する処理が考えられる．(c) データ出力処理とは，(b) のデータ加工処理が終了したデータを，出力あるいは一時保管用ファイルに保存することであり，たとえば日々の購入金額を該当月の総購入金額を保存しておく購入額データ・ベースに出力する処理が考えられる．

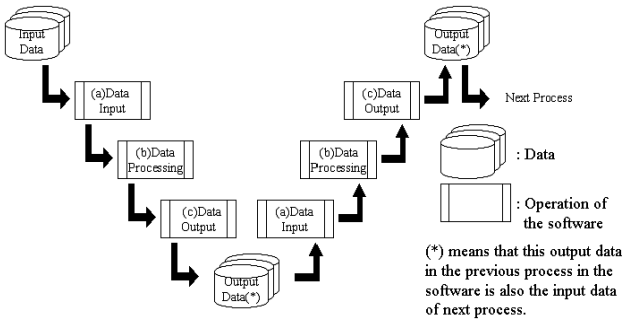

Fig. 3 Processing flow of SOIS

3.1.2 Windows 系システム開発ツールで開発された SOIS の特徴

ここでは，Windows 系システム開発ツール自体の特徴と，それを用いて開発された SOIS の特徴について記述する．

a. Windows 系システム開発ツールの特徴

- Windows オペレーティング・システム上で稼動する．
- 豊富なソフトウェア部品が準備されており，それを組み合わせて画面や処理を効率的に開発することができる．
- ソフトウェア部品を独自に作成，あるいは購入し，それらを組み合わせてシステムを開発することが可能である．
- 従来の手続き型処理に加えて，ソフトウェア部品の操作（クリック，ダブルクリックなど）時に関連付けておいた処理を実行することができる（イベント駆動型処理）．
- 他のビジネス・アプリケーションシステムと連携するために，豊富なインターフェース・モジュールが準備されている．

b. Windows 系システム開発ツールを用いて開発された SOIS の特徴

- 開発ツールを用いて開発された豊富な画面や印刷帳票を備えている．
- SOIS の１つの画面が，１つの事務処理の操作に対応しており，画面の遷移にしたがって事務処理が実行される．
- キー・ボード，マウス，ペンなどの様々な入力装置を使用することができ，データや操作指示の入力を容易に行なうことができる．

3.2 SOIS の開発工数の見積り手法

本節では，システム提案書作成段階における SOIS の開発工数見積り手法の仮説を記述する．

一般にソフトウェアの開発工数はソフトウェアの LOC

(Lines Of Code) と時間当たりの LOC 生産量を用いて，(1) 式のように表される．

$$MH = \frac{TLOC}{P} \quad (1)$$

MH： 開発工数
TLOC： ソフトウェアの LOC
P： 時間当たりの LOC 生産量

しかし，(1) 式をシステム提案書作成段階における SOIS の開発工数見積りに適用するには，以下の問題がある．

［問題 1］：SOIS 構成部位毎にアルゴリズムの複雑さの差異があり，それを開発する難易度にも差異が生ずるため，P を予測することが困難であること．

［問題 2］：Windows 系システム開発ツールを用いているため，多数のソフトウェア部品が含まれているが，それらを LOC に換算することが困難であること．

［問題 3］：システム提案書の記述内容から，SOIS の TLOC を直接予測することが困難であること．

［問題 1］を解決するためには，SOIS をサブ・システムに分割し，その単位で処理アルゴリズムの難易度を設定し，P を補正することが考えられる．

［問題 2］を解決するためには，使用するソフトウェア部品の LOC 換算率を設定し，標準的な LOC に置き換えることが考えられる．

［問題 3］を解決するためには，システム提案書の記述内容から，すべてのサブ・システム（機能）を漏れなく抽出し，その結果に基づいてサブ・システム毎の LOC を見積り，合算することで TLOC を求めることが考えられる．

その結果，(1) 式は (2) 式のように表すことができる．

$$MH = \sum_i \frac{SLOC_i \times CCR_i}{P_i} \quad (2)$$

ただし，

P_i： サブ・システム毎の時間当たりの LOC 生産量
CCR_i： サブ・システム毎に使用するソフトウェア部品の LOC 補正率
$SLOC_i$： サブ・システム毎の LOC

以降で，(2) 式で用いた P_i，CCR_i，$SLOC_i$ について検討する．

(1) サブ・システム毎の時間当たりの LOC 生産性 (P_i)

SOIS の処理は，社内業務を元の流れのままシステム化したものである．そのため，数値計算のような複雑なアルゴリズムが必要となる処理は存在しない．この結果，サブ・システム毎のアルゴリズムの複雑さに差異はないと考えられる．

同程度のアルゴリズムの複雑さを有するサブ・システムであれば，同レベルの技術力を有するシステム開発者による P は同程度になる．一般に，企業には様々な技術力のシステム開発者がいるが，同程度の重要度を持つソフトウェア開発であれば，全体として同程度の技術力を有する開発チームが組織される（経営的観点から言えば，経営者は，特別な要件が無ければ，どのようなソフトウェア開発でも同様な利益が生まれるよう，技術力が同程度になるよう開発チームを組織する）ので，P_i の差異はないと考えられる．したがって，P_i を一定値 P_{const} とみなす．

(2) サブ・システム毎のソフトウェア部品の使用に伴う LOC 補正率 (CCR_i)

SOIS の基本処理はデータ入力，データ加工，データ出力から構成されている．各サブ・システムの詳細な内容は対象となる事務処理で様々であるが，その流れは前述のように同様である．したがって，使用されるソフトウェア部品の種類と出現頻度は一定とみなして良く（たとえば，基本処理のデータ入力部分では，対象データは様々であるが，データ・ベース・アクセス・コントロールが，1 回使用される），基本処理毎の CCR_i の差異は生じない．したがって，CCR_i を一定値 CCR_{const} とみなす．

(3) サブ・システム毎の LOC($SLOC_i$)

上述の (1) から (3) の結果，(2) 式は (3) 式のように表すことができる．

$$MH = \sum_i \frac{SLOC_i \times CCR_{const}}{P_{const}}$$

$$= Const\_1 \times \sum_i SLOC_i \equiv Const\_1 \times TLOC \quad (3)$$

ただし，

$$Const\_1 = \frac{CCR_{const}}{P_{const}}$$

実際の SOIS は，複数の入力データと出力データを有しており，処理内容が複雑であるため，$SLOC_i$ の見積りは困難なものとなっている．加えて，システム提案書の記述内容からすべてのサブ・システムを抽出し，さらにその機能を漏れなく明確化することも困難である．そこで，$TLOC$ を見積るために，サブ・システムに代わって，基本処理（SOIS の処理内容を標準化したものであり (3.1) で詳述する）を導入し，$TLOC$ を見積る方法について検討する．以降に，基本処理の考え方，SOIS を構成する基本処理の個数，基本処理の LOC，それらを用いた SOIS の $TLOC$ の見積りについて記述する

(3.1) 基本処理の考え方

Fig.4 の上側に，SOIS の一例を示す．3.1(1) で述べたように SOIS の処理は，データの入力・加工・出力を繰り返しながら進められるが，通常，その処理の際に同時に使用される入力データは 2 個であり，出力データは 1 個となっている．これは，従来，事務処理システムは，2 個の入力データをつき合わせて，1 個の新しいデータ（中間または出力データ）を作成する処理を繰り返しながら処理を進めていく構成で設計されていた名残によるものだと考えられる．そのため，SOIS の処理も同様な構成となっていると考えられる．したがって，本論文では **Fig.4** の下側に示す 2 個の入力データと 1 個の出力データを持つ処理を基本処理と定義し，これを用いて SOIS

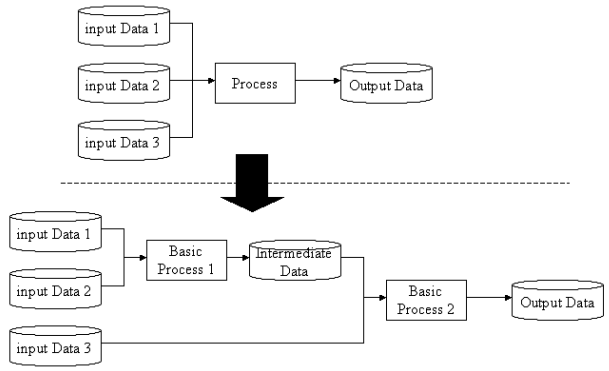

Fig. 4 Concept of basic process

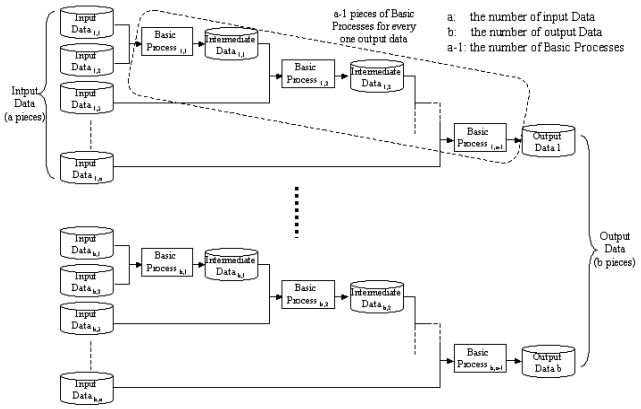

Fig. 5 Relationship between SOIS and basic processes

の処理を記述する．

(3.2) SOIS を構成する基本処理の個数

a 個の入力データと b 個の出力データを持つ SOIS を，基本処理を用いて記述した結果を，**Fig.5** に示す．この場合，1 個の出力データを計算するには，$(a-1)$[個] の基本処理が必要となる．ここで, 出力データを計算するには, すべての入力データが使用され，一種類の入力データに関わる計算は該当する基本処理の 1 箇所でまとめて行われるものと仮定する．したがって，a[個] の入力データと b[個] の出力データを有する SOIS の基本処理の個数は，$(a-1) \times b$[個] で抑えることができる．

(3.3) 基本処理の LOC

前述のように基本処理の内容は，2[個] のデータを入力し（データ入力），それらのデータ加工を行い（データ加工），そして 1[個] のデータを出力する（データ出力）ことである．基本処理で取り扱う入力データや出力データの規模は様々であるが，事務処理の場合には，すべての入力データに対して，同様の入力，加工，出力を繰り返すことになる．この繰り返し部分は Loop 命令などにより実現される．したがって，入力，加工，出力の処理の LOC は取り扱うデータの規模に依存しない．加えて，入力，加工，出力の内容は様々であるが，事務処理という業務内容の特徴上，複雑なものでなく LOC の差異は少ない．したがって，個々の基本処理の LOC（$BLOC$）は，ほぼ同じとなる．以上の結果，$BLOC$ を一定値 $BLOC_{const}$ とみなすことができる．

(3.4) SOIS の $TLOC$

(a) から (c) の結果，a 個の入力データと b 個の出力データを有する SOIS の $TLOC$ は，(4) 式で表すことができる．

$$TLOC = BLOC_{const} \times (a-1) \times b$$
$$= BLOC_{const} \times (ab-b) \quad (4)$$

ここで，今回対象としている SOIS の入出力データの個数は，3.4 節で後述するように，$16 \leqq a \leqq 83$ および $6 \leqq b \leqq 35$ である．したがって，(4) 式の計算結果は，括弧内の第 1 項の部分が支配的となり，(5) 式で近似することができる．

$$TLOC \cong BLOC_{const} \times ab \quad (5)$$

以上の検討の結果，SOIS の開発工数 MH は (6) 式で記述することができる．

$$MH \cong Const\_1 \times BLOC_{const} \times ab = Const\_2 \times ab \quad (6)$$

ただし，

$$Const\_2 = \frac{CCR_{const} \times BLOC_{const}}{P_{const}}$$

3.3 キーワード辞書

本節では，(5) 式の変数である入力データの個数 a と出力データの個数 b をシステム提案書から算出する方法ついて検討する．通常，これらの個数は，システム提案書には記述されていないため，以下の手法で算出する．

システム提案書の記述内容と入出力データの個数の関係を求めるため，既存のシステム提案書とソース・コードの対応について調査した．調査は，経験豊富なシステム開発者が，システム提案書のどの記述が，SOIS のどの入出力データに対応しているか，手作業で追跡することにより実施した．その結果，システム提案書のなかに入出力データの存在を表すキーワードが存在することが分かった．

たとえば，システム提案書に物品庫という記述がある場合，ソース・コードには物品マスタ，入庫マスタという 2 種類の入力データと，在庫マスタ，出庫マスタという 2 種類の出力データが存在する．これは，システム提案書に物品庫という記述が存在した場合，上記の入出力データと下記の処理が必要であることを意味している．

- 物品庫に保管されている物品の種類は物品マスタで管理されており，入力データとなっている．
- 物品庫に入庫される物品の個数は入庫マスタで管理されており，入力データとなっている．入庫した場合には在庫マスタを更新して最新の物品の在庫状態を把握できるようになっている．在庫マスタは出力データとなっている．
- 物品庫から出庫される物品は出庫マスタで管理されており，出庫した場合には在庫マスタを更新して，最新の物品の在庫状態を把握できるようにする．出庫マスタは出力デー

タとなっている．

このように SOIS の入出力データと関連のあるキーワードと入力データとの対応関係は 1 対 1 対応または 1 対多の対応となる．キーワードと入出力データの対応は **Table 1** に示すキーワード辞書で管理されている．キーワード辞書では，現在までに判明しているすべてのキーワードが登録されており，さらにそのキーワードに対応する入出力データが登録されている．このようにキーワードと入出力データの対応関係が明確に決まる理由は，対象が，同一システム開発企業が開発した SOIS という類似システムに限定されており，システム提案書の記述方法が暗黙のうちにルール化されたためだと考えられる．

SOIS の開発工数を見積る際には，はじめに，システム提案書からキーワードを抽出する．つぎに，キーワードと入出力データの関係を用いてキーワード毎の入出力データ数を求める．最後に入力データ数の総和を求める．ただし，重複して抽出されたキーワードは 1 回のみカウントするものとする．これは，基本処理が重複してカウントされることを防止するためである．以降，特に断らない限り，入出力データの個数は，この条件のもとでカウントするものとする．入力データ数と出力データ数は (7) 式および (8) 式で記述できる．

$$a = \sum_{i=1}^{n} num\_in(W_i) \tag{7}$$

$$b = \sum_{i=1}^{n} num\_out(W_i) \tag{8}$$

ただし，

$num\_in(W_i)$： キーワードに対応する入力データ数
$num\_out(W_i)$： キーワードに対応する出力データ数
W_i： 入出力データ関連ワード
n： 入出力データ関連ワードの数

(7) 式と (8) 式より，開発工数の見積り (9) 式が導かれる．

$$MH = Const\_2 \times (\sum_{i=1}^{n} num\_in(W_i)) \times (\sum_{i=1}^{n} num\_out(W_i)) \tag{9}$$

3.4 SOIS の開発工数見積り式の導出

SOIS の開発実績データを統計処理して，(6) 式と (9) 式の係数を求めると共に妥当性を確認する．このために，40 件の開発実績データを収集した．**Table 2** に，これらの SOIS のキーワード数・入力データ数・開発工数の実績を示す [3]．

以下に開発工数見積り式の導出手順を記述する．

(1) **Table 2** に示す SOIS のシステム提案書とソース・コードを比較して，キーワードと入力データとの対応関係を明らかにし，**Table 1** に示すキーワード辞書を作成する．キーワードの抽出と対応する入出力データの洗い出しは，経験豊富な SOIS 開発者が，手作業で行った．キーワード辞書に登録したキーワードの数は 118[個] であった．

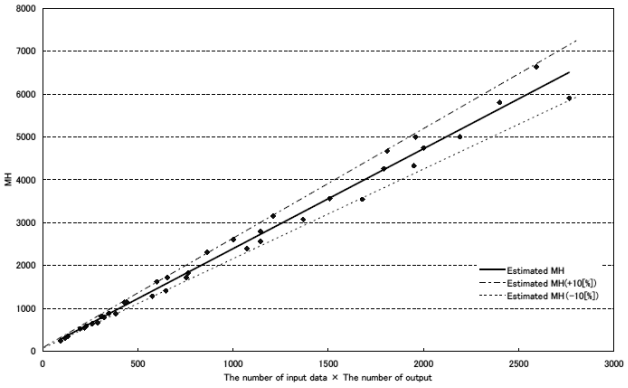

Fig. 6 Differnce between actual and estimated MH

(2) **Table 2** に示す SOIS のシステム提案書のなかからキーワードを抽出し，キーワード辞書を用いて，入力データの個数と出力データの個数を求める．結果を，**Table 2** の入出力データの個数の列に示す．

(3) 入力データと出力データの個数の積と開発工数の関係について，回帰分析を実施して，入力データの個数と出力データの個数の積 と MH の関係を求めた．

$$MH = 2.3ab + 67.5 \tag{10}$$

なお，MH と ab の決定係数は 0.99 であった．したがって，MH は ab で，良く説明できる．この結果は，3.2 節で述べた SOIS の開発工数は入力データの個数と出力データの個数の積に比例するという仮説と良く一致している．さらに，**Table 2** に示した実績の開発工数と (10) 式を用いて計算した見積り工数との誤差は，**Table 2** と **Fig.6** に示すように，ほぼ 10[%] 以内となった．なお，見積り工数との誤差は (11) 式で算出した．

$$EE = \frac{100 \times (AMH - EMH)}{AMH} \tag{11}$$

ただし，

EE： 見積り誤差
AMH：実績の開発工数
EMH：見積りの開発工数

以上の結果，SOIS の開発工数は入力データの個数と出力データの個数の積に比例すると考えることができ，(10) 式は 3.2 節で記述した開発工数見積りの仮説を良く説明できることが確認できた．

4. 提案する開発工数見積り手法の適用と評価

4.1 節で，提案手法を SOIS の開発工数見積りに適用した場合の結果について記述し，4.2 節で，その結果を評価する．

4.1 提案手法を適用するシステムの概要

Table 3 に示す 10 件の開発実績データを用いて提案手法を評価する．以下に，各 SOIS の概要を説明する．

SOIS − A は工事の予定工数を見積るシステムであり，

Table 1 Keyword dictionary - abridgment-

No	Keyword	Corresponding input data	Number of input data	Corresponding output data	Number of output data
1	物品倉庫	物品マスタ, 入庫マスタ	2	在庫マスタ，出庫マスタ	2
2	物品調達	物品マスタ，価格マスタ，購買先マスタ	3	物品マスタ	1
3	旅費精算	支払先マスタ，運賃マスタ	2	支払マスタ	2
4	勤怠管理	カレンダマスタ，勤怠マスタ，休暇マスタ	3	休暇マスタ，勤怠マスタ	2
5	購買処理	購買票，購買マスタ，部門費マスタ	3	支払票，部門費マスタ	2
- -	- - -	- - -	- - -	- - -	- - -

Table 2 Actual MH data for developing SOIS

SOIS	Number of Keywords	Number of input data	Number of output data	Actual MH[hours]	Estimated MH[hours]	Error[%]
1	5	16	6	240	229	4.8
2	6	17	7	312	283	9.2
3	6	19	7	351	317	9.8
4	8	22	9	525	471	10.2
5	8	23	10	588	548	6.9
6	9	22	10	547	524	4.2
7	9	24	13	809	742	8.2
8	9	24	12	663	686	-3.4
9	10	26	10	645	619	4.0
10	11	27	12	789	771	2.2
11	11	28	11	814	733	9.9
12	11	29	12	881	829	5.9
13	12	32	12	879	914	-4.0
14	13	33	13	1140	1021	10.4
15	14	34	13	1147	1052	8.2
16	14	36	16	1283	1371	-6.9
17	16	38	17	1412	1538	-8.9
18	17	40	15	1620	1429	11.8
19	17	41	16	1715	1561	-8.9
20	18	42	18	1723	1800	-4.5
21	18	45	17	1823	1821	0.1
22	19	48	18	2312	2057	11.0
23	21	50	20	2602	2381	8.5
24	21	51	21	2392	2550	-6.6
25	22	52	22	2792	2724	2.4
26	22	52	22	2559	2724	-6.4
27	23	55	22	3152	2881	8.6
28	23	57	24	3076	3257	-5.9
29	24	58	26	3561	3591	-0.8
30	27	60	28	3544	4000	-12.9
31	28	64	28	4258	4267	-0.2
32	23	67	27	4663	4307	7.6
33	29	69	29	4743	4764	-0.6
34	30	70	28	4998	4667	6.6
35	30	65	30	4329	4643	-7.3
36	31	73	30	5003	5214	-4.2
37	33	75	32	5805	5714	1.6
38	35	79	35	5901	6583	-11.6
39	35	81	32	6629	6172	6.9
40	37	83	34	6803	6663	2.1

SOIS － B は過去の工事の実績データを収集するシステムである．両システムの主なキーワードは工事番号，開発実績など，入力データは工事番号マスタ，開発実績マスタなど，出力データは見積りマスタ，開発実績マスタなどである．

SOIS － C は特許の申請状況を提供するシステムであり，特許の一覧と申請書類を表示するものである．主なキーワードは出願者，特許，出願年月日など，入力データは従業員マスタ，部門マスタ，特許出願マスタ，特許書類マスタなど，出力データは審査状況マスタ，特許マスタなどである．

SOIS － D は従業員の旅費の仮払いをするシステムであり，SOIS － E は従業員の旅費精算を管理するシステムである．主なキーワードは出張，行程，旅費など，主な入力データは

Table 3 Actual MH data for developing SOIS

No.	Number of Keywords	Number of input data	Number of output data	Actual MH[hours]	Estimated MH[hours]	Error[%]
A	11	32	13	986	1033	-4.8
B	13	38	14	1422	1304	8.2
C	16	61	25	3893	3628	6.8
D	16	24	12	647	733	-8.9
E	22	27	15	1089	1088	7.5
F	24	57	20	2601	2728	-4.9
G	24	60	22	3287	3149	4.2
H	26	49	18	2315	2124	8.3
I	28	52	24	2783	2980	-7.1
J	34	73	30	5009	5185	-3.5

出張経路データ，出張日時データ，従業員マスタ，手当マスタなど，出力データは仮払い指示票，旅費清算書などである．

SOIS － F は物品の発注を行なうシステムであり，SOIS － G は物品の受注を行なうシステムである．主なキーワードは物品，物品個数，納期，発注，受注など，入力データは部門マスタ，物品マスタ，在庫マスタ，発注先マスタ，受注元マスタなど，出力データは在庫マスタ，発注指示書，受注報告書，発注マスタ，受注マスタなどである．

SOIS － H は従業員の勤怠を管理するシステムである．主なキーワードは従業員，出退勤などであり，入力データは従業員マスタ，勤務記録，カレンダマスタなど，出力データは休暇マスタ，勤怠マスタなどである．

SOIS － I は部門の費用の計画をするシステムであり，SOIS － J は部門の費用の消費実績を管理するシステムである．主なキーワードは部門，費用計画，費用使用実績などであり，入力データは費目マスタ，使用実績，部門費用使用実績マスタなど，出力データは部門費計画マスタ，部門費用使用実績マスタなどである．

Table 3 の見積り開発工数の列に SOIS － A から SOIS － J に対して，(10) 式を用いて計算した見積り開発工数を示す．さらに，見積り開発工数と実績開発工数の差異を，**Table 3** の誤差の列に示す．**Table 3** より，提案手法を用いて SOIS 開発工数の見積った場合，その誤差は‐ 8.9[%] から＋ 8.2[%] であり，平均は 5.3[%] であった．

4. 2 提案手法の適用結果の評価

Table 3 より，提案手法を用いた場合の見積り開発工数の誤差は± 10[%] 以内であり，精度が高いといえる．

小規模システム開発企業の経営の観点から，開発工数見積りを見ると，実際の開発工数が見積った開発工数を下回った場合は経営に与えるリスクは小さく，上回った場合はリスクは大きいといえる．そこで，提案手法を適用した事例のなかで，実績工数が見積り工数を比較的大きく上回った SOIS － B，SOIS － E，SOIS － H について，システム提案書と完成ソフトウェアの対応関係について調査した．その結果，それぞれ 2 個，1 個，2 個のキーワード辞書に登録されていない出力データが存在することが分かった．これは，事前に想定していない新しい処理が，それぞれの SOIS に含まれていたことを意味している．提案手法の場合，開発工数は「入力データの個数」と「出力データの個数」の積に比例すると考えるため，キーワード辞書に登録されていない出力データが存在したことにより，該当する処理部分の開発工数が見積りから落ちてしまったためだと考えられる．しかしながら，出力データがキーワード辞書に登録されていないことに起因する開発工数の見積り誤差は，高々，2〜3[%] である．この数値は SOIS の工数見積りにおいて誤差として十分に許容できる範囲である．これは SOIS ドメインが，成熟しているため想定していない新しい処理が追加されることが，非常に稀であるためだと考えられる．

そのため，このように 1〜2[個] のキーワードが未登録となることがあるが，このことが提案手法に与える影響は小さいと考えられる（ただし，適時，キーワード辞書と (10) 式の係数の更新し，最新の状態を維持することは必須である）．なお，実際の開発工数が見積りの開発工数を下回った場合でも，その差は 9.0[%] 以下であり，十分に許容可能な誤差の範囲内にあるということができる．

以上の結果，提案手法を適用して，システム提案書の作成段階で開発工数を精度良く見積ることができ，その適用を成功させるためには，「同一のシステム開発企業」，「類似した SOIS の開発工数見積り」という条件のもとで，「システム提案書のなかに SOIS で実施する処理に関係するキーワードが漏れなく記述されていること」，「システム提案書からキーワードの抽出が漏れなく行われていること」，「関係するキーワードがキーワード辞書に漏れなく登録されていること」「最新の開発工数の実績を用いて，常に開発工数の見積り式を適切な状態に維持しておくこと」が必須だといえる．

最後に，経営的観点から提案手法の評価を行なう．提案手法を SOIS の開発工数見積りに適用した場合，ほぼ± 10[%] の誤差で開発工数を見積ることが可能である．実際にシステム開発企業からシステム発注企業に提示される価格は開発工数に換算して，10[%] 程度の安全率と 10〜20[%] 程度の利益を含んでいる．そのため，(10) 式を用いて SOIS の開発工数を見積った場合，見積り開発工数の誤差は前述の安全率のなかにあり，さらに利益まで含めて考えれば，受注した SOIS の開発工数が不足するリスクは非常に低いと考えられる．

以上の結果，提案手法は，SOIS のシステム提案書の作成段階における有効な開発工数見積り手法であると考えることができ，提案手法を用いて開発工数を見積ることで，開発工数の不足により，受注した開発が赤字となるリスクは非常に低いということができる．

5. 終わりに

本論文では，SOIS の開発工数を，そのシステム提案書の記述内容から，誤差 10[%] 程度で見積る手法を提案した．提案手法を使用することで，システム提案書作成段階において開発工数を精度良く見積ることが可能となり，開発の期間や費用を容易に見積ることが可能になった．

この結果，システム開発企業にとって，利益の計画と管理が容易になるとともに，システム提案書作成段階における開発費用や開発期間の見積りに関するリスクを低減することが可能になると考えられる．

今後は，キーワード辞書の充実を図り，SOIS の開発工数の見積り (10) 式をより精度の高いものにしていく．さらに，異なるシステム開発企業，および，システム分野における開発工数見積りへの適用を行い，提案手法の評価を深めていく．

謝辞　NEDO 研究開発事業「次世代輸送系システム設計基盤技術開発プロジェクト」のなかで精度の高いソフトウェアの開発工数の見積り手法を研究しました．本論文は，この研究成果の一部を SOIS の開発工数見積りに適用した結果について報告したものです．この場を借りて関係各位にお礼を述べさせていただきます．

参 考 文 献

1）内田: すぐわかる EXCEL による多変量解析, 東京図書 (1996)
2）小谷: ファンクション・ポイント法によるソフトウェア開発規模・工数見積の現状, 関西情報センター (1998)
3）齋藤: 開発規模見積もりのためにソフトウエアドキュメントに関する研究, 筑波大学大学院ビジネス科学研究科経営システム専攻修士論文 (2004)
4）永井: RFP &提案書完全マニュアル, 日経 BP(2005)
5）Boehm B.: Software Engineering Economics, Prentice Hall(1981)
6）Hamphery W.: A Discipline For Software Engineering, Addison-Wesley Publishing Company, 105/109(1995)
7）IEEE: IEEE Guide to Software Requirement Specifications, IEEE(1984)
8）Jones C.: Estimating Software Costs, McGraw-Hill Companies(1998)
9）Putnam L. and Myers W.: Measures for Excellence: Reliable Software on Time, within Budget, Yourdon Press(1992)
10）Sprouls J.: IFPUG Function Point Counting Practice Manual, Release 3.0, International Function Point Users Group(1990)
11）Takahashi M., Terano T. and Tsuda K.: Estimation of Lines of Codes and Execution Time of Embedded System, International Conference of Emergent Synthesis, 187/192(2001)
12）Dolado J.: A Validation of the Component-Based Method for Software Size Estimation, IEEE Transactions on Software Engineering, **26**-10, 1006/1021(2000)
13）Matson J., Barret B,, and Melichamp J.: Software Development Cost Estimation Using Function Points, IEEE Transactions on Software Engineering, **20**-4, 275/287(1994)
14）Costagliola G., Ferrucci F.,Tortora G. and Vitiello G.: Class Point: An Approach for the Size Estimation of Object-Oriented Systems, IEEE Transactions on Software Engineering, **31**-1, 52/74(2005)
15）Auer M., Trendowicz A., Graser B., Haunschmid E. and Biffl S.: Optimal Project Feature Weights in Analogy-Based Cost Estimation: Improvement and Limitations, IEEE Transactions on Software Engineering, **32**-2, 83/92(2006)
16）Jørgensen M.: Evidence-Based Guidelines for Assessment of Software Development Cost Uncertainty, IEEE Transactions on Software Engineering, **31**-11, 942/954(2005)
17）Porter-Roth B.: Request for Proposal - A Guide to Effective RFP Development -, Addison-Wesley(2002)
18）真野, 誉田: 見積りの方法—ソフトウェア開発における実践的見積り技法—, 日科技連 (1993)
19）Horowitz E., Madachy R., Steece B., Clark B. and Bohem B: Software Cost Estimation with COCOMO , Prentice-Hall(2000)

［著 者 紹 介］

高 橋 正 和（正会員）

1988 年 立教大学理学部物理学科卒業．同年石川島播磨重工業 (株) 入社．2002 年 (株) ギャラクシーエクスプレス出向．1998 年 筑波大学 大学院 経営システム科学専攻修了．2002 年 同大学院博士課程修了．2005 年 島根大学 総合理工学部数理・情報システム学科 助教授，(株) ギャラクシーエクスプレス 顧問．博士（システムズマネジメント）．情報処理学会，電気学会等会員．

齋 藤 達 也

1997 年 産能大学 経営情報学部 情報学科卒業．同年 (株) インテック入社．2004 年 筑波大学 大学院 ビジネス科学研究科 経営システム科学専攻修了．修士（MBA）．

福 江 義 則

1990 年 関西学院大学 経済学部卒業．同年 富士通株式会社入社．2004 年 筑波大学 大学院 ビジネス科学研究科 経営システム科学専攻修了．2005 年 徳島大学 大学院 工学研究科 情報システム工学専攻入学．情報処理学会，人工知能学会の会員．

津 田 和 彦

1986 年 徳島大学 工学部 情報工学科卒業．同年 三菱電機（株）入社，1991 年 住友金属工業（株）入社．1994 年徳島大学 大学院 工学研究科システム工学専攻修了．工学博士．1998 年 筑波大学 大学院 経営システム科学専攻 助教授．2005 年 同教授．情報処理学会，電子情報通信学会等会員．