

画面仕様書に基づく対話型ソフトウェアの 複雑度重みつき機能規模の測定技法

野中 誠[†] 角 頼 章 広^{††}
ブカーリ イサム^{††} 東 基 衛^{†††}

本論文では、対話型ソフトウェアの複雑度重みつき機能規模を、画面仕様書に基づいて測定する技法を提案する。提案技法では、まず、複雑度重みのない機能規模（標準機能規模と呼ぶ）を、画面構成部品を分類してその中から機能部品を識別し、これにデータ移動サブプロセスを割り当てることにより測定する。次に、複雑度属性に基づいて各データ移動サブプロセス型を分類し、各分類に複雑度重みを与える。最後に、この重みを用いて複雑度重みつき機能規模を測定する。対話型ソフトウェアの開発では画面仕様書が主な要求仕様であり、開発初期に得られることが多い。提案技法は標準機能規模および複雑度重みつき機能規模を画面仕様書から直接的に測定するため、測定効率の向上や開発工数の早期見積りが期待できる。提案技法によって測定された標準機能規模は、COSMIC-FFP 測定法によって測定された機能規模とほぼ同等である。そのため、提案技法を開発初期に用いて COSMIC-FFP 測定法による測定値を予測することで、本技法を COSMIC-FFP 測定法の補完的技法として用いることができる。提案技法をカーナビゲーション製品の対話部ソフトウェアに適用し、標準機能規模および複雑度重みつき機能規模を測定した。測定結果と論理ソースコード行数（SLOC: Source Lines of Code）との相関を分析した結果、複雑度重みつき機能規模により SLOC の予測性が高まることが確認できた。

A Method for Measuring Complexity-weighted Functional Size of Interactive Software Using User Interface Specifications

MAKOTO NONAKA,[†] AKIHIRO KAKURAI,^{††} ESSAM BUKHARY^{††}
and MOTOEI AZUMA^{†††}

This paper proposes a method for measuring complexity-weighted functional size of interactive software using user interface specifications. The functional size without complexity weight (called 'standard functional size') is measured by identifying functional components from user interface specifications and assigning data movement sub-processes to the identified components. The complexity weight is assigned to each classification of each data movement sub-process type. By using the weights, the complexity-weighted functional size is measured. User interface specifications can be obtained at earlier stages of development process of interactive software, as the specifications are usually regarded as major requirements specifications. By measuring functional size using the specifications directly, not only measurement efficiency improvement but early effort estimation can be expected. The proposed method can be used as the complement of the COSMIC-FFP measurement method, as the standard functional size is nearly equal to the COSMIC-FFP functional size. The proposed method was applied to the interactive part of an actual car-navigation software system in order to validate its SLOC (Source Lines of Code) predictability. The result showed that the complexity-weighted functional size, measured by the proposed method, could be a good predictor of SLOC.

[†] 早稲田大学理工学総合研究センター

Advanced Research Institute for Science and Engineering, Waseda University

^{††} 早稲田大学大学院理工学研究科経営システム工学専門分野

Graduate School of Science and Engineering, Waseda University

^{†††} 早稲田大学理工学部経営システム工学科

School of Science and Engineering, Waseda University

1. はじめに

近年の携帯電話やカーナビゲーション製品などの対話型システムは、利用者要求の高度化にともなってソフトウェア規模が増大しつつある。一方、新製品を市場に少しでも早く出荷するために、開発プロジェクトには納期短縮が求められている。これらの制約を同時

に満たすためには、要求定義段階でソフトウェアの機能規模を測定し、開発工数およびコストを見積もり、顧客と開発者の合意に基づいた開発計画を立てることが必要である。

ソフトウェア機能規模の測定法には、開発の早期に利用可能であること、測定値が利用者の観点から意味のある値として理解できること、および工数見積りに役立つことが求められる。ソフトウェア機能規模測定の方法は ISO/IEC 14143-1¹⁾に規定されており、これに適合した測定法の 1 つに COSMIC-FFP (Common Software Measurement International Consortium - Full Function Point) 測定法²⁾がある。機能規模を測定するにあたって、このような標準的な測定法を用いることは、複数ソフトウェア間の機能規模やコストの比較を行う際に有用である。

COSMIC-FFP 測定法では、利用者機能要件に基づいて抽象ソフトウェアモデルを作成し、このモデルに基づいて機能規模を測定する。しかし一般に、対話型ソフトウェアの開発では画面仕様書が主な要求仕様であり、これが開発初期に得られることが多い。したがって、画面仕様書から機能規模を直接的に測定することができれば、測定効率の向上や開発工数の早期見積りが期待できる。

また、COSMIC-FFP 測定法は機能規模の測定法であるため、複雑度は考慮されていない。しかし開発工数をより高い精度で見積もるためには、各機能単位の複雑度を考慮した機能規模を測定し、これを用いて開発工数を見積もる必要がある。

本研究の目的は、対話型ソフトウェアを対象として、複雑度重みのない機能規模(これを標準機能規模と呼ぶ)および複雑度重みつき機能規模を効率的に測定する技法を開発することである。提案技法によって測定される標準機能規模は、COSMIC-FFP 測定法によって測定される機能規模とほぼ同等の値となるようにする。これにより、提案技法を COSMIC-FFP 測定法の補完的測定法として用いることができ、機能規模を効率的に測定し、開発プロセスの初期に開発工数を見積もることが期待できる。

本論文では、対話型ソフトウェアの複雑度重みつき機能規模を、画面仕様書に基づいて測定する技法を提案する。提案技法をカーナビゲーション製品の対話部ソフトウェアに適用し、画面仕様書から複雑度重みつき機能規模の測定を行った。測定された値と論理ソースコード行数(SLOC: Source Lines of Code)との相関を分析し、複雑度重みつき機能規模による SLOC の予測性を評価した。

本論文の構成は以下のとおりである。2 章では、従来の機能規模および複雑性の測定に関する研究とその問題点を述べる。3 章では、本研究における複雑度重みつき機能規模測定のアプローチを述べる。4 章では画面仕様書から標準機能規模を測定する技法を、5 章では複雑度重みの決定方法および複雑度重みつき機能規模の測定方法について述べる。6 章では提案技法を実際の対話型ソフトウェアに適用し、複雑度重みつき機能規模の測定を行った結果と、SLOC との相関を分析した結果を述べる。測定結果と提案技法に関する考察を 7 章で述べ、最後に 8 章で本論文のまとめを述べる。

2. 従来研究

この章では、機能規模および複雑度の測定に関する従来研究とその問題点について述べる。

2.1 機能規模と複雑性

機能規模の測定とは、被測定ソフトウェアが提供する機能の量を、利用者の観点から測定することである。ISO/IEC 14143-1 では、機能規模を「利用者機能要件を定量化して得られるソフトウェアの規模」と定義している。要求仕様に対して適用できる機能規模の測定技法には、IFPUG (International Function Point Users Group) 法³⁾や COSMIC-FFP 測定法など様々な技法があげられる。

複雑性は、IEEE STD 729-1983 では「システムまたはシステム構成要素の複雑さの度合い」と定義されている。システム全体に関する複雑度には、IFPUG 法の一般システム特性などがあげられる。また、システム構成要素に関する複雑度では、ソースコードを対象とした McCabe のサイクロマティック数⁴⁾、要求仕様を対象とした IFPUG 法の複雑度重み、Tran-Cao らが提案した COSMIC-FFP 測定法の複雑度メトリクス⁵⁾などがあげられる。

2.2 IFPUG 法

IFPUG 法は、Albrecht らが提案したファンクションポイント (FP: Function Point) 法⁶⁾を洗練させたものである。IFPUG 法では、内部論理ファイル、外部インタフェースファイル、外部入力、外部出力および外部照会の 5 種類の機能要素について、複雑度の重みを加えたうえで合計し、未調整 FP を算出する。各機能要素の複雑度は、参照するファイル数やデータ項目数などによって、「低い」「普通」および「高い」の 3 段階により与えられる。複雑度重みの値は、「低い」と「高い」との間で約 2 倍異なる。こうして得られた未調整 FP に 14 個の一般システム特性の評定値を反

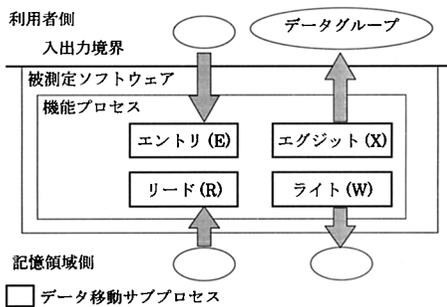
図1 COSMIC-FFP 測定法での測定対象要素²⁾

Fig. 1 Elements to be measured by the COSMIC-FFP measurement method.

映させ、最終FPを得る。

IFPUG法は広く利用されているが、機能規模のほかに品質要求や技術要求などが一般システム特性として扱われているため、機能規模のみを測定するという観点からは問題点が指摘されている⁷⁾。このように、測定法によって機能規模の概念が異なることは問題であるため、ISO/IEC 14143-1において機能規模の基本概念が規定されている。

2.3 COSMIC-FFP 測定法

COSMIC-FFP測定法は、アプリケーションソフトウェアやリアルタイムソフトウェアなどIFPUG法よりも広い領域に対して適用性の高い機能規模測定法であり、ISO/IEC 14143-1に適合した測定法である。機能規模を測定するにあたって、被測定ソフトウェアに含まれる機能プロセスと、利用者側および記憶領域側にあるデータグループを識別する。機能プロセスは、利用者側および記憶領域側とソフトウェアとの間でデータグループを移動させる処理から構成される。このような処理をデータ移動サブプロセスと呼び、リード(R)、ライト(W)、エントリ(E)およびエグジット(X)の4種類に分類される。これらの関係を図1に示す。

これらのデータ移動サブプロセスのうちEとXに関しては、利用者に対するものと他サブシステムに対するものに分けられる。本論文ではこれらを明確に区別し、利用者に対するEおよびXをそれぞれ E_u および X_u 、他サブシステムに対するEおよびXをそれぞれ E_s および X_s と表記する。

COSMIC-FFP測定法では、利用者機能要件に基づいて、機能プロセス、データグループおよびデータ移動サブプロセスを識別することで抽象ソフトウェアモデルを作成し、このモデルを用いて機能規模を測定する。識別されたデータ移動サブプロセスを計数した値が被

測定ソフトウェアの機能規模となり、Cfsu (Cosmic Functional Size Unit) という単位で表される。

COSMIC-FFP測定法によって測定された機能規模と開発工数との間には、十分に検証された結果ではないものの、高い相関があることが報告されている^{8),9)}。しかし、開発工数をより高い精度で見積もるためには、機能単位の複雑度を測定し、複雑度重みつき機能規模を用いて開発工数を見積もる必要がある。

Tran-Caoらは、COSMIC-FFP測定法に関する複雑度メトリクスを提案している⁵⁾。これは、機能プロセスに対する複雑度であり、機能要素の最小単位であるデータ移動サブプロセスに対する複雑度ではない。機能プロセスの機能規模は、含まれるデータ移動サブプロセスの数によって異なる。Tran-Caoらの複雑度メトリクスは、機能規模で正規化された値ではないため、複雑度重みつき機能規模を算出する際に不整合が生じうる。IFPUG法でも複雑度重みが機能要素の最小単位に与えられているように、COSMIC-FFP測定法においてはデータ移動サブプロセスに対して複雑度重みが与えられる方が望ましい。

2.4 仕様書に基づく機能規模測定

柏本らは、イベントトレース図に基づいてIFPUG法のFP値を自動計測する手法を提案している¹¹⁾。またDiabらは、リアルタイムアプリケーション向けオブジェクト指向モデル化言語ROOM (Realtime Object-Oriented Modeling) に基づいて、COSMIC-FFP機能規模を測定する方法を示している¹²⁾。しかし、これらは比較的詳細に定義された仕様に対してのみ適用できる手法であるため、開発初期における開発工数見積りに適用することは困難である。

3. 研究アプローチ

本論文では、対話型ソフトウェアの複雑度重みつき機能規模を、画面仕様書に基づいて測定する技法を提案する。以下に、本研究のアプローチを述べる。

3.1 提案技法の測定対象要素

提案技法における標準機能規模の測定対象要素を図2に示す。提案技法では、画面仕様書の対話画面で識別できる機能部品に基づいて、利用者与被測定ソフトウェア間のデータ移動に関連するデータ移動サブプロセスを測定する。ただし、COSMIC-FFP測定法では、対話画面の機能部品と関連しないデータ移動(図2点線部)も測定対象としている。提案技法では、このようなデータ移動サブプロセスを測定対象としない。

しかし、対話型ソフトウェアの場合、多くのデータ

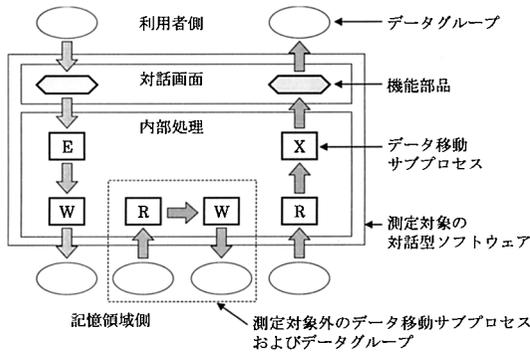


図2 提案技法での測定対象要素

Fig. 2 Elements to be measured by the proposed method.

移動は利用者と被測定ソフトウェア間で行われると考えられる。したがって、提案技法における標準機能規模の測定値と COSMIC-FFP 測定法の測定値との差異は小さいと考えられる。そのため、機能規模の測定に COSMIC-FFP 測定法を用いる際に、提案技法を補完的に用いることができる。すなわち、開発初期に提案技法を用いることで、COSMIC-FFP 測定法によって測定された機能規模 (COSMIC-FFP 機能規模と呼ぶ) を予測することができる。

提案技法における標準機能規模は COSMIC-FFP 機能規模と完全に一致するものではない。したがって、COSMIC-FFP 測定法の機能規模の単位を表す Cfsu と区別するために、本技法で測定された機能規模は Cfsu' という単位で表す。

3.2 標準機能規模の測定

提案技法では、画面仕様書から標準機能規模を測定する方法として、画面仕様書に含まれる画面構成部品を分類し、その中から機能部品を識別し、機能部品にデータ移動サブプロセスを割り当てる。ここで画面構成部品とは、一般にウィジェットと呼ばれるような対話画面を構成している個々の部品であり、たとえばボタン、ラベル、文字列入力領域などがあげられる。また機能部品とは、利用者にとって意味のあるまとまりのデータを入出力するのに用いられる部品であり、複数の画面構成部品から構成される。これらの詳細は 4.1 節で述べる。

提案技法では、COSMIC-FFP 測定法のように抽象ソフトウェアモデルを作成することなく、画面仕様書から直接的に標準機能規模を測定する。そのため、測定効率の向上が期待できる。

3.3 複雑度重みの決定

本研究では、2.3 節で述べたデータ移動サブプロセス型について、画面構成部品の分類および部品数に基

づいて複雑度に関する分類を行い、複雑度重みを決定する。この重みを用いて、複雑度重みつき機能規模を測定する。詳細は 5 章で述べる。

データ移動サブプロセス型の複雑度に関する分類は、プログラミング言語などのソフトウェア実装環境に依存せずに行うことができる。一方、複雑度重みは、実装環境によって異なる値が割り当てられる。Jones は、1FP あたりの平均ソースコード行数を様々なプログラミング言語について示している¹³⁾。提案技法に対してもこれと同様に、1 データ移動サブプロセスあたりの複雑度重みを、様々な実装環境に対して与えることができる。

3.4 提案技法の適用対象

提案技法は、COSMIC-FFP 測定法に基づいた測定技法である。したがって、COSMIC-FFP 測定法が適用対象外としているソフトウェアは、提案技法もまた対象外となる。文献 2) によると、COSMIC-FFP 測定法では以下の領域のソフトウェアを適用対象外としている。

- (1) 複雑な計算アルゴリズムや特殊で複雑な規則を特徴とするソフトウェア。
- (2) 連続的に変数処理するソフトウェア。

たとえば、カーナビゲーション製品の対話部ソフトウェアでは、地図データを連続的に処理して利用者に表示する機能を有する。また携帯電話でも動画像を扱う機能が存在する。これらの機能は前述の適用対象外の条件に合致する。したがって、提案技法を用いて対話型ソフトウェアの機能規模を測定する際には、これらの機能を別サブシステムとして分離し、被測定ソフトウェアの範囲外とするか、異なる測定技法を適用する必要がある。

以上の理由から、提案技法は、複雑な計算アルゴリズムや連続的な変数の処理を含まない対話型ソフトウェアに適用するものとする。

4. 標準機能規模の測定

この章では、画面仕様書に基づいて標準機能規模を測定する技法について述べる。

4.1 画面構成部品の分類

画面構成部品を、入力部品、出力部品、複合部品、スクロール部品および制御部品の 5 種類に分類する。このうち、入力部品、出力部品および複合部品を機能部品と呼び、これらにデータ移動サブプロセスを割り当てる。各分類について以下に説明する。

(1) 入力部品

入力部品とは、利用者からのデータ入力を受け付け

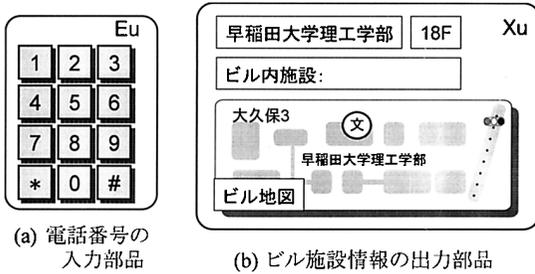


図3 機能部品の例

Fig. 3 Examples of functional components.

る機能を提供する画面構成部品群である。入力部品を構成する画面構成部品の例として、ボタンや文字列入力領域などがあげられる。入力部品は、利用者から1つのデータグループをソフトウェア内に取り込む機能を提供するため、1つのEuを割り当てることができる。また、その入力部品によって取り込まれたデータグループが、他サブシステムまたは記憶領域に渡される場合は、それぞれ1つのXsまたは1つのWがさらに割り当てられる。

たとえば、図3(a)の電話番号入力用の入力部品は、1つの入力部品が10個の数字ボタンと2つの記号ボタンから構成される。この例のように、利用者にとって意味のある1つのデータグループを入力する機能単位をもって、1つの入力部品とする。

(2) 出力部品

出力部品とは、利用者に対してデータを出力する機能を提供する画面構成部品群である。出力部品を構成する画面構成部品の例として、文字列、画像などがあげられる。出力部品は、利用者に対して1つのデータグループを出力する機能を提供するため、1つのXuが割り当てられる。また、その出力部品によって出力されるデータグループが、他サブシステムまたは記憶領域から取り込まれる場合は、それぞれ1つのEsまたは1つのRがさらに割り当てられる。

図3(b)の例では、ビル施設情報という1つのデータグループが4つの画面構成部品、すなわち3つの文字列と1つの画像によって表示されている。このように、利用者にとって意味のある1つのデータグループを出力する機能単位をもって、1つの出力部品とする。

(3) 複合部品

入力部品と出力部品の2つの機能をあわせ持った機能部品を複合部品と呼ぶ。複合部品を構成する画面構成部品の例として、ボタンラベルが計算結果などによって変化するボタンなどがあげられる。複合部品には、1つのEuと1つのXuが割り当てられる。入力部品

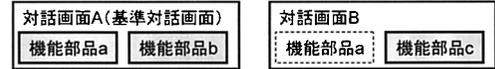


図4 基準対話画面に対する機能部品の割当ての例

Fig. 4 Assigning functional components to a base interactive screen.

や出力部品と同様、Xs、W、EsおよびRが割り当てられる場合がある。

(4) スクロール部品

スクロール部品は、機能部品が扱うデータグループに対して、スクロール機能を付加する部品である。この部品はスクロールという機能を提供していると考えられることができるため、機能規模の測定対象としてとらえることもできる。しかし提案技法では、COSMIC-FFP測定法と同様、データ移動にのみ着目して機能を測定するため、スクロール部品は機能部品とせず、データ移動サブプロセスを割り当てない。

(5) 制御部品

制御部品は、画面遷移制御を行う部品である。スクロール部品と同様、制御部品はデータグループの移動をとまわらないため、データ移動サブプロセスを割り当てない。

4.2 基準対話画面への機能部品の割当て

同一の機能を提供する機能部品が、複数の対話画面に含まれている場合がある。たとえばカーナビゲーション製品の場合、地図描画機能は多くの対話画面中で共通して提供される機能である。これらの機能を個別に測定すると、1つの機能を重複して測定することになり、正確な機能規模が得られない。

この問題を避けるために、複数の対話画面に同一の機能を提供する機能部品がある場合は、ある1つの基準対話画面に対してのみその機能部品を割り当てる。図4の例では、機能部品aが対話画面AおよびBに含まれている。この場合、対話画面Aを基準対話画面とし、機能部品aを対話画面Aにのみ割り当てる。

4.3 測定手順

提案技法における標準機能規模の測定手順を以下に述べる。

Step 1) データグループの識別

個々の対話画面について、対話画面と、利用者、他サブシステムおよび記憶領域との間で移動するデータグループを識別する。データグループの候補は、4.1節で述べた機能部品の粒度に対応したものを識別する。

Step 2) 機能部品の識別とデータ移動サブプロセスの割当て

4.1節で述べた分類に基づいて画面構成部品を分類し、

データグループ 機能部品	利用者			他サブシステム		記憶領域
	データA データB ...	Eu Eu	データA データB	
入力部品	部品A	Eu		Xs		
	部品B	Eu	...	Xs
	部品C	Eu		Xs		...
出力部品	部品D	Xu		Es		
	データ移動サブ プロセス集計	E, R X, W	Eu Eu Xu	Es Xs Xs		
機能規模合計		3		3		0

図5 標準機能規模測定テンプレートと記入例

Fig. 5 Standard functional size measurement template.

識別されたデータグループに対応する機能部品を識別する。同一の機能を提供する機能部品が複数の対話画面に含まれる場合は、1つの対話画面を基準対話画面として、その機能部品を基準対話画面にのみ割り当てる。識別された機能部品には、4.1節に基づいてデータ移動サブプロセスを割り当てる。

Step 3) 標準機能規模の測定

以上の方法で識別されたデータ移動サブプロセスを数し、対話画面ごとの標準機能規模を測定する。測定には、図5の標準機能規模測定テンプレートを用いる。1つの対話画面において識別されたデータグループおよび機能部品を、図5の表頭および表側にそれぞれ列挙する。これらの交点に、識別されたデータ移動サブプロセス型の記号を記入する。ここで、同一のデータグループに対して、同一のイベントによって実行される機能部品が存在した場合は、同じ列に記入する。最後に、縦方向にデータ移動サブプロセスを集計する。

各対話画面の標準機能規模は、式(1)により算出する。

$$FSS_j = \sum_{DMSP_{ij} \in FSS_j} Size(DMSP_{ij}) \quad (1)$$

ここで、 FSS_j は対話画面 j の標準機能規模、 $DMSP_{ij}$ は対話画面 j で識別された i 番目のデータ移動サブプロセスを表す。関数 $Size(DMSP_{ij})$ の値は1である。

測定された各対話画面の標準機能規模を合計し、被測定ソフトウェア全体の標準機能規模 FS を式(2)で算出する。

$$FS = \sum_j FSS_j \quad (2)$$

4.4 標準機能規模の測定例

図6に示したカーナビゲーション製品の対話画面例を対象に、標準機能規模の測定例を示す。

図6の対話画面において、データグループとして「ルート詳細データ(IC名, 距離, 料金, および到着



図6 対話画面の例

Fig. 6 A sample interactive screen.

時刻が示された行の集合)」が識別できる。入力部品として地図ボタン群が識別できる。図6の対話画面上には3個の地図ボタンが示されているが、これらはルート詳細データに対して共通の操作を提供するため、1つの入力部品として識別する。この入力部品は、ルート詳細データの中から利用者が選択した1行を受け取り(Eu),これを他サブシステムに出力する(Xs)。また、出力部品としてルート詳細データを表示する画面構成部品群が識別できる。これは、他サブシステムからルート詳細データを受け取り(Es),利用者へ出力する(Xu)。RETURNボタンは制御部品、スクロールバーおよび上下矢印ボタンはスクロール部品である。したがって、この対話画面では4つのデータ移動サブプロセスが識別され、標準機能規模は4Cfsu'となる。

5. 複雑度重みの決定と複雑度重みつき機能規模の測定

この章では、各データ移動サブプロセス型の分類に対する複雑度重みを、画面仕様書に基づいて決定する方法について述べる。また、複雑度重みつき機能規模の算出方法を述べる。

5.1 データ移動サブプロセス型の分類

4.1節で述べた画面構成部品の分類により、Eu, Xu, Es, Xs, RおよびWの6種類のデータ移動サブプロセス型が識別される。これらについて、画面仕様書から測定可能な属性に基づき、複雑度に関する分類を行う。以下に、各データ移動サブプロセス型に対する分類を示す。

(1) ユーザからのエントリ(Eu)

Euの複雑度は、ユーザからデータを受け取る機能に必要な処理量と考えられる。これは、ボタンや文字列入力領域などの入力部品の種類、入力部品を構成する画面構成部品の数、および入力部品に関連するスクロール部品などによって異なると考えられる。したがって、これらをEuの複雑度に関する属性とする。

図3(a)のEuの例では、入力部品の種類はボタン、構

成する画面構成部品数は 12 個，関連するスクロール部品なし，と識別できる．

(2) ユーザへのエグジット (Xu)

Eu の場合と同様，出力部品の種類，出力部品を構成する画面構成部品の数，および出力部品に関連したスクロール部品を，Xu の複雑度に関する属性とする．図 3 (b) の Xu の例では，出力部品の種類は文字列と画像であり，構成する画面構成部品数はそれぞれ 3 個と 1 個，関連するスクロール部品なし，と識別できる．

(3) 他サブシステムからのエントリ (Es)

Es の複雑度は，他サブシステムからデータを読み込む機能に必要な処理量と考えられる．これは，他サブシステムから読み込むデータ型およびパラメータ数に依存すると考えられる．したがって，データグループに含まれるデータ属性の型および数を，Es の複雑度に関する属性とする．

(4) 他サブシステムへのエグジット (Xs)

Xs の複雑度は，他サブシステムに対してデータを出力する機能に必要な処理量と考えられる．これは，Es と同様に，他サブシステムに渡すデータ型およびパラメータ数に依存すると考えられる．また，他サブシステムが提供する API (Application Programming Interface) の呼び出し数や，API 呼び出し時における条件分岐数も複雑度に関する属性として扱うことができる．ただし，API に関する属性は開発初期の画面仕様書では明示されないことが多いため，つねに利用できる属性とは限らない．

(5) リード (R)

R の複雑度は，記憶領域からデータを読み込む機能に必要な処理量と考えられる．これは Es の複雑度と同様であるため，R の複雑度に関する属性は Es と同様とする．つまり，上記 (3) で述べた属性を R についても測定する．

(6) ライト (W)

W の複雑度は，記憶領域に対してデータを出力する機能に必要な処理量と考えられる．これは Xs の複雑度と同様であるため，W の複雑度に関する属性は Xs と同様とする．つまり，上記 (4) で述べた属性を W についても測定する．

5.2 複雑度重みの決定

5.1 節で述べた各データ移動サブプロセス型の分類に対して複雑度重みを決定する．複雑度重みは，SLOC や開発工数の予測のために用いる値であるため，プログラミング言語や開発ツールなどの実装環境によって値が異なる．したがって，測定対象の実装環境に特化した値を決定する．

複雑度重みを与える方法として，以下の 3 種類の方法があげられる．

(1) 開発工数に基づく重みづけ

(2) ソースコードから測定可能な規模および複雑度メトリクスに基づく重みづけ

(3) 主観評価に基づく重みづけ

このうち，(1) の開発工数に基づいて複雑度重みを与える場合は，各データ移動サブプロセスの作成にかかった工数を測定する必要がある．しかしこのような精密な工数の測定は現実的には困難であり，画面単位あるいはソースコードファイル単位で開発工数を測定し，その値から各データ移動サブプロセスの工数を推定するのが現実的な方法である．(2) は，SLOC または McCabe のサイクロマチック数などの値に基づいて複雑度重みを与える方法である．これは，各データ移動サブプロセスを実装しているソースコードが正確に追跡できる場合に適用できる．(3) は，各データ移動サブプロセス型分類について，各分類の実現上の困難度合いを主観的に比較することで複雑度重みを与える方法である．具体的には，AHP (Analytical Hierarchy Process)⁴⁾ の適用が考えられる．

これらの方法のいずれかを適用することで，実装環境に固有の複雑度重みを算出する．

5.3 データ移動サブプロセス型の複雑度分類例

図 6 の対話画面例を対象に，データ移動サブプロセス型の複雑度の分類例を示す．

図 6 の対話画面において，入力部品である地図ボタンから識別された Eu は「1 つの固定ボタンから構成された Eu」に分類できる．これは，図 3 (a) のように「複数の固定ボタンから構成される Eu」に比べて複雑度が低いと考えられる．

ルート詳細情報から識別される Xu は，IC 名，距離，料金，アイコンおよび到着時刻といった複数のデータ属性を出力する．また，このデータグループの表示にはスクロール機能が付与されている．したがってこの Xu は「複数のデータ属性から構成されるスクロール機能つき Xu」に分類できる．これは「単一のデータ属性から構成される Xu」に比べて複雑度が高いと考えられる．

5.4 複雑度重みつき機能規模の測定

複雑度重みを決定した後に，式 (3) によって画面ごとの複雑度重みつき機能規模を算出する．

$$WFSS_j = \sum_{DMSP_{ij} \in FSS_j} W(DMSP_{ij}) \quad (3)$$

ここで， $WFSS_j$ は対話画面 j の複雑度重みつき機能規模， $W(DMSP_{ij})$ は対話画面 j に含まれる i 番

目のデータ移動サブプロセスに与えられた複雑度重みを算出する関数である。複雑度重みつき機能規模の尺度を表すのに $W-Cfsu'$ を用いる。

被測定ソフトウェア全体の重みつき機能規模 WFS は、式 (4) で求められる。

$$WFS = \sum_j WFSS_j \quad (4)$$

6. 適用事例

提案技法を実際のソフトウェア製品に適用し、標準機能規模および複雑度重みつき機能規模の測定を行った。以下にその結果を述べる。

6.1 標準機能規模の測定

某社カーナビゲーション製品の対話部ソフトウェアを対象に、提案技法を適用して標準機能規模を測定した。対話部ソフトウェアは、総対話画面数が 170 画面、SLOC が 78,739 行という規模のソフトウェア製品であった。全画面のうち、3.4 節に述べた理由により、提案技法の適用範囲外と判断した 14 画面を除いた 156 画面について、画面あたり標準機能規模を手作業で測定した。

測定対象の画面仕様書からは、各機能部品について、 E_s , X_s , R および W のいずれのデータ移動サブプロセスが含まれているかの識別が行えなかった。しかし測定対象の対話部ソフトウェアは、入力部品によって利用者から取り込まれたデータを、他のサブシステムに出力する機能を必ず有しているものであったため、1 入力部品に $1E_u$ と $1X_s$ を割り当てた。同様に、出力部品についても $1E_s$ と $1X_u$ を割り当てた。したがって、各機能部品の標準機能規模の値は偶数となり、各画面の標準機能規模も偶数となっている。

図 7 に、画面あたり標準機能規模の分布を示す。図 7 の横軸は各階級における標準機能規模の代表値を、縦軸は各階級に含まれる画面数を表す。共通機能を基準画面に割り当てて測定した結果、機能部品が割り当てられなかった画面が 26 画面存在した。これらの画面を差し引いたため、図 7 における画面数合計は 130 に減少している。これら 130 画面の標準機能規模の平均は 5.4 $Cfsu'$ 、標準機能規模の合計は 700 $Cfsu'$ であった。8 $Cfsu'$ 以下の画面は全体の 90.8% を占めており、ほとんどの画面がこの範囲に含まれていた。また、標準機能規模が 22 $Cfsu'$ の画面が 1 つ観測されたが、これは基準画面に設定された画面であり、複数画面に含まれる機能部品をこの画面の機能として測定したため、標準機能規模が大きくなっている。

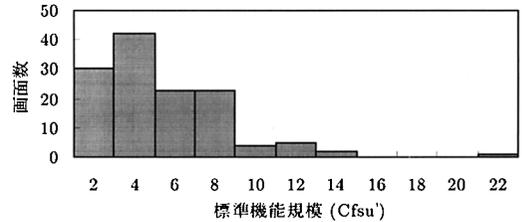


図 7 画面あたり標準機能規模の分布
Fig. 7 Distribution of standard functional size per screen.

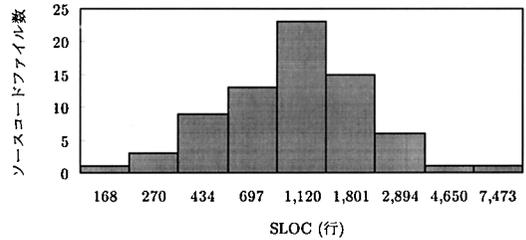


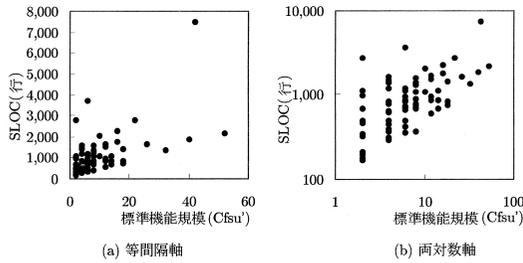
図 8 ソースコードファイルあたり SLOC の対数分布
Fig. 8 Logarithmic distribution of SLOC per source code file.

6.2 標準機能規模と SLOC との相関分析

提案技法によって測定された標準機能規模が、実績規模である SLOC の予測に有効な尺度となるかを検証するために、標準機能規模と SLOC との相関分析を行った。測定した各ソースコードファイルの SLOC の対数分布を図 8 に示す。図 8 の横軸は各階級における SLOC の上限値を、縦軸は各階級に含まれるソースコードファイル数を表す。ソースコードファイル数は 72 個、ソースコードファイルあたりの SLOC の対数平均は 865.7 行であった。ここで SLOC の対数平均とは、各ソースコードファイルの SLOC の対数値の平均を計算し、その平均値で自然対数の底 e のべき乗を計算した値である。したがって SLOC の対数平均の単位には、SLOC と同様に「行」を用いている。

被測定ソフトウェアには、1 画面が必ずしも 1 ソースコードファイルとして実装されておらず、複数画面が 1 ソースコードファイルに実装されたものが多数含まれていた。このようなソースコードファイルの SLOC を画面単位に明確に区別することは困難であるため、各ソースコードファイルの SLOC と、そのファイルによって実装された対話画面の標準機能規模合計との相関を分析した。

散布図を図 9 に示す。図 9 (a) は両軸とも等間隔の値を用いた散布図、図 9 (b) は両対数軸の散布図である。相関係数はそれぞれ 0.574 と 0.598 であった。無相関の検定¹⁵⁾を行った結果、いずれも有意水準 5% で無相関ではないことが確認できた(棄却域



(a) 等間隔軸 (b) 両対数軸

図9 標準機能規模とSLOCの散布図

Fig. 9 Scatter diagrams between standard functional size and SLOC.

$r(70, 0.05) = 0.2319$). ただしいずれも強い相関があるとはいえない。

図7および図8から読みとれるように、標準機能規模およびSLOCは小さな値に多くのデータが集中している。したがって、図9(b)の両対数の散布図を用いた方が、散布状態の把握およびSLOCの予測には有用と思われる。

6.3 複雑度重みの決定

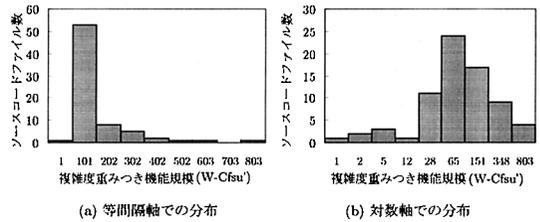
SLOCの予測性を高めるために、各機能部品を構成する部品数を用いて複雑度重みつき機能規模を測定する。それに先立ち、複雑度重みの値を決定した。5章で述べたとおり、提案技法ではデータ移動サブプロセス型に対して複雑度に関する分類を行う。しかし6.1節で述べたように、測定対象ソフトウェアでは、画面仕様書からはEs, Xs, RおよびWのデータ移動サブプロセスの識別が十分に行えなかった。そのため、機能部品に対して複雑度に関する分類を行った。

入力部品の複雑度分類には、入力部品を構成する画面構成部品の数を用いた。具体的には、カーナビゲーションシステムの対話部ソフトウェアという製品領域の性質上、入力部品はすべてボタンで実現されていたため、ボタンの数を用いた。出力部品についても、画面構成部品の数を用いた。具体的には、外部サブシステムからデータを読み込んで表示する際に用いられるデータ部品、すなわち文字列可変ラベル、可変アイコン、グラフィックおよび計算値を表示する部品数を用いた。複合部品についても画面構成部品の数、具体的にはラベル可変ボタンの数を用いた。

複雑度重みの決定は、ソースコードファイル全72ファイルを対象として、対話画面の重みつき機能規模とSLOCとの両対数軸における相関係数がおおむね最大値となるような複雑度重みを探索的に決定した。こうして得られた機能部品を構成する画面構成部品1個あたりの複雑度重みを表1に示す。この値は今回の測定対象ソフトウェアの実装環境に固有の値である。

表1 構成部品1個あたりの複雑度重み
Table 1 Complexity weights per user interface component.

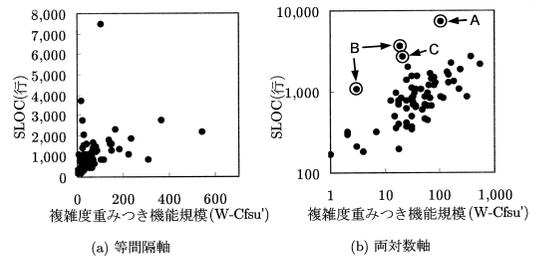
機能部品	部品1個あたりの複雑度重み
入力部品	6.0
出力部品	1.0
複合部品	3.0



(a) 等間隔軸での分布 (b) 対数軸での分布

図10 画面あたり複雑度重みつき機能規模の分布

Fig. 10 Distribution of complexity-weighted functional size per screen.



(a) 等間隔軸 (b) 両対数軸

図11 複雑度重みつき機能規模とSLOCとの散布図

Fig. 11 Scatter diagrams between complexity-weighted functional size and SLOC.

6.4 複雑度重みつき機能規模の測定

表1の複雑度重みと式(3)を用いて、各対話画面の複雑度重みつき機能規模を算出した。図10に、画面あたりの複雑度重みつき機能規模の分布を示す。図10(a)は横軸の機能規模を等間隔の値で、図10(b)は対数軸で表している。横軸の数値は、各階級の上限值を表している。複雑度重みつき機能規模の合計は4,846 W-Cfsu' となった。

6.5 複雑度重みつき機能規模とSLOCとの相関分析

複雑度重みつき機能規模とSLOCとの散布図を図11に示す。図11(a)は両軸とも等間隔の値を用いた散布図、図11(b)は両対数の散布図である。相関係数はそれぞれ0.342と0.618であった。無相関の検定を行った結果、有意水準5%でいずれも無相関ではないことが確認できた(棄却域は6.2節の値と同じ)。

等間隔軸の散布図、図11(a)は6.2節で示した等間隔の散布図、図9(a)よりも相関係数が小さくなった

ものの、両対数の散布図、図 11 (b) に関しては図 9 (b) よりも相関係数が必然的に大きな値となった。しかし、図 11 (b) と図 9 (b) との差はほとんどない。母相関係数の同等性の検定を行った結果、有意水準 5% で両者の母相関係数が異なるとはいえないという結論が得られた (P 値 = 0.854)。 P 値がきわめて高いことから、両者の相関係数に差があるとは考えにくい。

ただし、図 11 (b) の点 A は、1 つのソースコードファイルで 17 画面を実装したものであり、また、地図上でのスクロール機能を有する複雑な画面が多く含まれていた。点 B は、機能部品が扱うデータグループのデータ属性数が、他の画面と比べて多かった。点 C は、データ属性の表示条件が複雑な画面が含まれており、また、機能規模の測定対象外であった画面のソースコードが含まれていた。

これら 4 点を除いた場合の相関係数は 0.742 であり、両者の間には強い相関があるといえる。この値は、図 9 (b) の相関係数よりも大きな値である。しかし、母相関係数の同等性の検定を行った結果、有意水準 5% では両者の母相関係数が異なるとはいえなかった (P 値 = 0.131)。ただし、 P 値は比較的小さな値であることと、実現上の制約にも大きく影響を受ける SLOC 尺度との相関を考えた場合、図 9 (b) における相関係数 0.598 から 0.742 に向上したことは技術的観点からは意味があることと考えられる。以上より、複雑度重みを与えることによって、SLOC の予測性が高まったと考えられる。

7. 考察および評価

以上の結果について考察を述べ、提案技法に関する評価を行う。

7.1 SLOC の予測性

両対数軸の散布図において 4 つの点を除外したうえで、複雑度重みつき機能規模と SLOC との間に強い相関がみられた。今回の分析では複雑度の属性として画面構成部品の数のみを用いたが、これらの属性によって SLOC の予測性が高まったといえる。また、データ属性数や画面の分類などといった他の属性を用いることで、さらに予測性を高めることができると思われる。一般的には SLOC と開発工数との間には相関があるので、複雑度重みつき機能規模によって開発工数の見積りも可能であると考えられる。

また、図 9 (b) において、標準機能規模に対する SLOC の下限がほぼ直線的に与えられている。このことから、ある標準機能規模が与えられたときに、その対話画面の実装に必要な SLOC の下限が予測でき

ると思われる。複雑度重みは、この下限 SLOC から乖離する SLOC の予測に用いるパラメータと見なすことができる。

ただし、共通する機能部品を基準対話画面に割り当てたものの、実装時には、非基準対話画面においてこれらの機能部品の実装に多くのソースコードを作成している場合がある。したがって、提案技法を SLOC の予測に特化して用いる場合は、共通する機能部品を個別に測定し、各対話画面に割り当てるなど、提案技法の修正が必要である。

なお、冗長なソースコード記述であれば、提供する機能量にかかわらず SLOC が増大する。また、リファクタリング¹⁶⁾のような、開発工数を費やして無駄な SLOC を減少させる活動を実施すると、開発工数と SLOC との相関は弱くなるか、あるいは無相関になりうる。今回の測定対象では各画面の開発工数が得られなかったため分析が行えなかったが、複雑度重みつき機能規模と開発工数との相関を分析し、開発工数の予測性について検討する必要がある。

7.2 複雑度重みの値

本論文では、複雑度重みつき機能規模と SLOC との全体的な相関係数を最大化することで、複雑度重みを決定した。各機能部品に対応した SLOC を個別に計測して複雑度重みを算出する方法も考えられるが、実際には関数モジュールどうしが複雑に呼び出しあっているため、各機能部品に関する SLOC を明確に区別することは困難であった。このように、機能部品間の複雑度重みに関する定量的な検証が困難であるため、複雑度重みの妥当性に関する定性的な考察を以下に述べる。

イベント駆動型ソフトウェアの場合、ボタン押下などのイベント発生時に、システム状態に基づいてデータ授受の処理を行う。特に対話型ソフトウェアの場合は、入力部品に関連したイベントが支配的であるため、入力部品の処理は一般に複雑となり SLOC が多くなる。データ取得および表示処理は、入力部品によって定義された処理に基づいて定期的に行われる場合が多いと考えられ、対応する SLOC は少ないと考えられる。なお、複合部品は両者の機能を有するため、両者の中間的な存在と考えられる。

このような観点から、少なくとも複雑度重みの大小の順序に関しては妥当であると考えられる。今後、複雑度重みの値の大きさに関する検証や、他の複雑度属性を利用した場合の重みの検討、あるいは開発環境の差異を考慮した重みの検討などを行う必要がある。

7.3 測定効率

提案技法は、画面仕様書から直接的に標準機能規模および複雑度重みつき機能規模を測定するものである。提案技法では抽象的なソフトウェアモデルを作成することなく測定できるため、測定効率は COSMIC-FFP 測定法に比べて高いと思われる。提案技法は COSMIC-FFP 測定法と補完的に用いることができる。すなわち、開発初期に提案技法を用いて機能規模を効率的に測定し、その後、仕様が詳細化された後に COSMIC-FFP 測定法を適用し、COSMIC-FFP 機能規模を測定することができる。

また、画面仕様書の作成時に、画面構成部品に対して、入力部品、出力部品または複合部品のいずれかの属性を与えることによって、標準機能規模および複雑度重みつき機能規模の測定をある程度自動化することが可能である。今回の測定では手作業で測定したが、このような自動化ルールを定義し、計測ツールを開発することによって、開発の早期に効率的に測定することが期待できる。

7.4 提案技法の対話型ソフトウェアへの適用妥当性

提案技法は、COSMIC-FFP 測定法に基づいた測定技法である。ここでは、COSMIC-FFP 測定法を対話型ソフトウェアに適用することの妥当性について確認することで、提案技法の対話型ソフトウェアに対する適用妥当性を考察する。

本研究では、対話型ソフトウェアとして、携帯電話やカーナビゲーションシステムなどの対話部ソフトウェアを対象としている。対話型ソフトウェアは、利用者からの入力を対話的に受け取り、これをソフトウェア内に取り込んで処理を行い、記憶領域にデータを保存する。また、システムイベントなどによって記憶領域からデータが読み出され、利用者に提供される。このように、対話型ソフトウェアはデータ移動が主となるシステムであり、機能規模の測定に COSMIC-FFP 測定法および提案技法を適用することは妥当であると考えられる。ただし、3.4 節で述べた適用対象を考慮する必要がある。

8. おわりに

本論文では、対話型ソフトウェアの標準機能規模および複雑度重みつき機能規模を、画面仕様書に基づいて測定する技法を提案した。また、提案技法を実際の対話型ソフトウェア製品に適用し、標準機能規模および複雑度重みつき機能規模の測定を行った。これらの測定結果の対数値と SLOC の対数値との相関を分析した結果、両者の間に正の相関が確認できた。外れ値

を除外すると、複雑度重みつき機能規模を用いることで SLOC の予測性はさらに高まることが確認できた。

今後の課題として、提案技法によって測定された複雑度重みつき機能規模が、SLOC だけでなく開発工数の予測に有用な尺度となることを確認する必要がある。また、複雑度の分類のさらなる検討や、複雑度重みつき機能規模の測定の自動化を検討する必要がある。

参考文献

- 1) ISO/IEC 14143-1: 1998, Information Technology — Software Measurement, Functional Size Measurement, Part1: Definition of Concepts (1998).
- 2) Abran, A., et al.: *COSMIC-FFP Measurement Manual*, ver.2.0, Common Software Measurement International Consortium (COSMIC) (1999). 小泉寿男ほか(訳): COSMIC-FFP 測定マニュアル 第 2.0 版 (2000).
- 3) IFPUG: Function Point Counting Practice Manual, Release 4.1 (1999).
- 4) McCabe, T.J.: A Complexity Measure, *IEEE Trans. Softw. Eng.*, Vol.SE-2, No.4 (1976).
- 5) Tran-Cao, D., Abran, A. and Levesque, G.: Functional Complexity Measurement, *11th International Workshop on Software Measurement (IWSM2001)* (2001).
- 6) Albrecht, A.J. and Gaffney, J.E.: Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation, *IEEE Trans. Softw. Eng.*, Vol.9, No.6, pp.639–648 (1983).
- 7) Fenton, N.E. and Pfleeger, S.L.: *Software Metrics: A Rigorous and Practical Approach*, 2nd ed., International Thomson Computer Press (1997).
- 8) Abran, A., Symons, C. and Oligny, S.: An Overview of COSMIC-FFP Field Trial Results, *12th European Software Control and Metrics (ESCOM2001)*, pp.277–285 (2001).
- 9) 長野伸一, 眞瀬健一, 渡邊保夫, 綿引隆一, 西山 茂: 交換機ソフトウェアへの機能規模測定法 (COSMIC-FFP) 適用と評価, 情報処理学会第 63 回全国大会講演論文集 (分冊 1), pp.141–142 (2001).
- 10) Boehm, B.W., et al.: *Software Cost Estimation with COCOMO II*, Prentice Hall PTR (2000).
- 11) 柏本隆志, 楠本真二, 井上克郎, 鈴木文音, 湯浦克彦, 津田道夫: イベントトレース図に基づく要求仕様書からのファンクションポイント計測手法, 情報処理学会論文誌, Vol.41, No.6, pp.1895–1904 (2000).

- 12) Diab, H., Frappier, M. and St-Denis, R.: A Formal Definition of COSMIC-FFP for Automated Measurement of ROOM Specifications (2001). <http://www.lrgl.uqam.ca/>
- 13) Jones, C.: *Applied Software Measurement: Assuring Productivity and Quality*, second edition, McGraw-Hill (1996). 鶴保征城ほか(監訳): ソフトウェア開発の定量化手法 第2版, 共立出版 (1998).
- 14) 木下栄蔵(編著): AHPの理論と実際, 日科技連 (2000).
- 15) 永田 靖: 入門 統計解析法, 日科技連 (1992).
- 16) Fowler, M.: *Refactoring: Improving the Design of Existing Code*, Addison-Wesley (2000). 児玉公信ほか(訳): リファクタリング プログラミングの体質改善テクニック, ピアソン・エデュケーション (2000).

(平成 14 年 4 月 10 日受付)

(平成 14 年 9 月 5 日採録)



野中 誠(正会員)

1995年早稲田大学理工学部卒業。1997年同大学大学院修了。2000年同大学院博士後期課程単位取得退学。1999年早稲田大学理工学部経営システム工学科助手。2002年より早稲田大学理工学研究センター嘱託、現在に至る。ソフトウェア工学、特にメトリクスおよび工数見積りに関する研究に従事。第61回全国大会奨励賞受賞、生涯教育委員会委員、情報規格調査会 SC7/WG12 小委員会委員、IEEE 会員。



角頼 章広(学生会員)

2002年早稲田大学理工学部卒業。現在、同大学大学院修士課程在籍。ソフトウェア工学に関する研究に従事。



Bukhary Essam

2002年早稲田大学理工学部卒業。現在、同大学大学院修士課程在籍。ソフトウェア工学に関する研究に従事。



東 基衛(正会員)

1963年早稲田大学第一理工学部卒業。同年日本電気(株)入社。1987年より早稲田大学理工学部経営システム工学科教授、現在に至る。この間、1991年より1992年まで英国 South Bank University 客員教授。ソフトウェア工学(特に品質管理、メトリクス、要求定義等)、ユーザインタフェース等の研究に従事。1996年情報規格調査会標準化功績賞受賞。編著書、監訳書等多数。ISO/IEC 9126 および 14598 シリーズチーフエディタ。ISO/IEC JTC1/SC7/WG6 Convenor, 情報規格調査会 SC7/WG6 小委員会主査。IEEE, ACM, 日本経営工学会, 経営情報学会, 人間工学会, 日本品質管理学会, 経営システム学会各会員。