

画面仕様書に基づく対話型ソフトウェアの 機能規模と複雑度の測定技法

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

COSMIC-FFP 測定法はソフトウェア機能規模測定法のひとつであり、IFPUG 法よりも幅広いソフトウェア製品領域に適用できる。本稿では、対話型ソフトウェアの COSMIC-FFP 機能規模および複雑度を、画面仕様書に基づいて測定する技法を提案する。機能規模の測定は、画面構成部品を分類して機能部品を識別し、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行う。複雑度は、複雑度に関する属性に基づいて各データ移動サブプロセス型を分類し、これに複雑度重みを与えることにより測定する。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、本技法を用いることで、機能規模の測定効率の向上や開発工数の早期見積りが期待できる。本稿では、提案技法の概要とその適用結果について報告する。

A Method for Measuring Functional Size and Complexity of Interactive Software based on Interface Specifications

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

The COSMIC-FFP measurement method is a method for measuring software functional size. In this paper, a method is proposed for measuring COSMIC-FFP functional size and complexity of interactive software based on interface specifications. The functional size is measured by classifying interface parts, identifying functional components and assigning data movement sub-processes to them. The complexity is measured by classifying each data movement sub-process type based on complexity attributes and assigning complexity weight to them.

Interface specifications can be obtained at earlier stages of software development process, as they are usually regarded as a major requirements specifications in interactive software development. The proposed method can be effective in improving the measurement efficiency and early effort estimation. The overview of the method and the preliminary result of its application is described in this paper.

1. はじめに

近年の携帯電話やカーナビゲーションシステムなどの対話型システムは、利用者要求の高度化に伴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および複雑度を測定し、開発工数を見積り、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、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と、工数見積りに有効であることが求められる。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ひとつに COSMIC-FFP (Common Software Measurement International Consortium -Full Function Point) 測定法²⁾ がある。この測定法は、IFPUG (International Function Point Users Group) 法³⁾ に比べて幅広い製品領域に適用できるように設計されている。

COSMIC-FFP 測定法では、利用者機能要件に基づいて抽象的なソフトウェアモデルを作成し、これ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

[†] 早稲田大学理工学部 経営システム工学科
Dept. of Industrial and Management Systems Eng.,
School of Science and Eng., Waseda University

り、これが開発初期に得られることが多い。画面仕様書から機能規模を直接的に測定することで、測定効率の向上や開発工数の早期見積りが期待できる。

COSMIC-FFP 測定法では機能的側面のみ測定され、複雑性は考慮されていない。しかし工数をより高い精度で見積るためには、機能規模の測定に加えて複雑度を測定し、工数の見積り値を調整する必要がある。したがって、COSMIC-FFP 測定法に対応した複雑度の測定技法が必要である。

本稿では、対話型ソフトウェアの COSMIC-FFP 機能規模および複雑度を、画面仕様書に基づいて測定する技法を提案する。また、提案技法を実際の対話型ソフトウェア製品の画面仕様書に適用し、機能規模の測定を行った。

本稿の構成は以下の通りである。2章では、従来の機能規模および複雑度測定に関する研究とその問題点を述べる。3章では、本研究における機能規模と複雑度測定のアプローチを述べ、対話型ソフトウェアに対して COSMIC-FFP 測定法を適用することの妥当性について述べる。4章では画面仕様書から機能規模を測定する技法を、5章では複雑度を測定する技法について述べる。6章では提案技法を実際の対話型ソフトウェア製品に適用し、機能規模を測定した結果を述べる。最後に7章で本稿のまとめと今後の課題を述べる。

2. 従来研究

2.1 機能規模と複雑性

機能規模の測定は、利用者の観点からソフトウェアが提供する機能の量を測定する。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法や COSMIC-FFP 法などがある。

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」と定義している。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挙げられる。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 のサイクロマチック数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⁾などが挙げられる。

2.2 IFPUG 法

IFPUG 法は、Albrecht 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

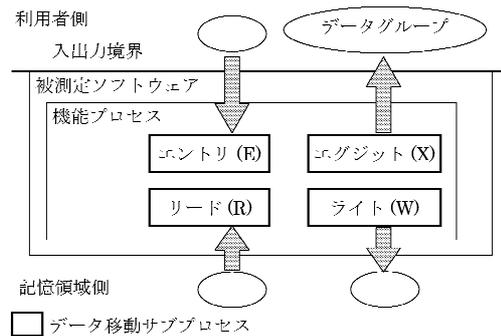


図1 COSMIC-FFP 測定法での測定対象要素²⁾

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、機能規模の他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など、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に適用可能な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트리 (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。このうち、E と X に関しては、ユーザ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。本稿ではこれらを明確に区別し、ユーザに対する 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、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、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るためには、各データ移動サブプロセスの複雑度を測定し、機能規模の値を調整したうえで開発工数を見積る必要がある。

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、粒度のやや大きい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。機能プロセスは、含まれるデータ移動サブプロセスの数によって機能規模が異なる。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論理的な不整合が生じうる。

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、機能規模を複雑度によって重みづけるには、データ移動サブプロセスに対する複雑度が定義されている必要がある。

2.4 仕様書に基づく機能規模測定

仕様書から機能規模を測定する研究も行われている。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

以上の従来研究および問題点に対して、本研究では、対話型ソフトウェアの COSMIC-FFP 機能規模および複雑度を、画面仕様書に基づいて測定する技法を提案する。以下に、機能規模および複雑度測定のアプローチと、対話型ソフトウェアに COSMIC-FFP 測定法を適用することの妥当性について述べる。

3.1 機能規模の測定

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。提案技法では、画面仕様書から直接的に測定できる要素を分類することで、COSMIC-FFP 機能規模を測定するアプローチをとる。

画面仕様書から機能規模を測定する方法として、本研究では、画面仕様書に含まれる画面構成部品を分類して機能部品を識別し、これにデータ移動サブプロセスを割り当てる。画面構成部品とは、対話画面を構成している個々の部品であり、例えばボタン、ラベル、文字列入力領域などが挙げられる。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的な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機能規模を測定する。その結果、測定効率の向上が期待できる。

3.2 複雑度の測定

本研究では、システム全体に対してではなく、COSMIC-FFP 測定法における構成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を測定する。機能規模の測定と同様、複雑度についても画面構成部品の分類に基づいて測定する。6 種類のデータ移動サブプロセス型について、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。各分類を構成する部品数に基づいて複雑度を測定する。測定された複雑度の値に重みを与えることで、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分類は、プログラミング言語等の実装環境に依存せずに行うことができる。しかし複雑度重みは、実装環境によって異なる値が割り当てられる。Jones は、1 FP 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。

3.3 COSMIC-FFP 測定法適用の妥当性

提案技法について述べる前に、COSMIC-FFP 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しておく必要がある。2.3 節で述べた通り、COSMIC-FFP 測定法はデータ移動に着目した機能規模測定法である。ただし、以下の領域は対象外としている。

- (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) 連続的に変数を処理するソフトウェア。

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等の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、システムイベント等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中心のシステムであり、機能規模の測定に COSMIC-FFP 測定法を適用することは妥当であると考えられる。

しかし、例えばカーナビゲーションシステムの場合は、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。このような機能は COSMIC-FFP 測定法による測定には適合しない。したがって、COSMIC-FFP 測定法で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、または異なる測定法を適用する必要がある。

以上の理由から、本稿で提案する測定技法は、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. 画面仕様書に基づく機能規模測定技法

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。以下に各分類について説明する。

(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挙げられる。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 E_u 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 X_s または 1 つの W がさらに割り当てられる。

例えば、図 2(a) の電話番号の入力部品は、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

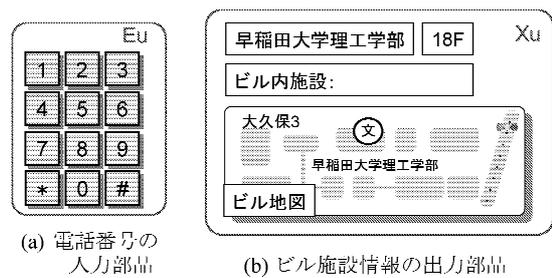


図 2 機能部品の例

る画面構成部品の例として、文字列、画像などが挙げられる。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 X_u 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 E_s または 1 つの R がさらに割り当てられる。

図 2(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 つの機能を併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挙げられる。複合部品には、1 つの E_u と 1 つの X_u が割り当てられる。入力部品や出力部品と同様、 X_s , W , E_s 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、入力部品および出力部品が表示する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ため、機能規模の測定対象として捉えることもできる。しかし COSMIC-FFP 測定法では、データ移動にのみ着目して機能を測定する。したがって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、画面遷移制御や操作指示を行う部品である。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伴わないため、データ移動サブプロセスを割り当てない。

4.2 機能プロセスと対話画面との対応づけ

提案技法では、1 つの対話画面を 1 つの機能プロセスに対応させる。この対応づけは、COSMIC-FFP 測

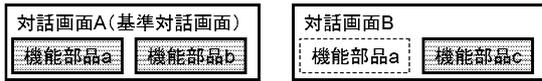


図 3 基準対話画面に対する機能部品の割り当ての例

定法の機能プロセスの定義と概念が、対話画面と一致している必要がある。

COSMIC-FFP 測定法では、以下の原則を満たしたものを機能プロセスとしている。

- (1) 少なくとも 1 つの利用者機能要件から導出。
- (2) 識別可能なイベントが発生したとき実行。
- (3) 少なくとも 2 つのデータ移動を含む。
- (4) ふたつ以上の待ち状態を含まない。

1 つの対話画面を 1 つの機能プロセスに対応させるには、対話画面がこれらの原則を満たしている必要がある。ここで、画面仕様書は利用者要求に基づいて作成されたものであるため、対話画面は利用者要求の基本単位と考えることができる。また、対話画面は、利用者操作またはシステム内で発生するイベントによって表示され実行される。さらに、対話画面はユーザ入力受付という 1 つの待ち状態にある。以上から、(1)、(2) および (4) の原則を満たしている。また (3) の原則を満たすために、機能プロセスとして識別する対話画面を、ユーザ入力を記憶領域または他サブシステムへと移動させる、またはその逆の機能を有したものに限定する必要がある。

なお、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。例えばカーナビゲーションシステムの場合、現在地の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、複数の対話画面に共通する機能部品があり、それらが同一の機能を提供する場合は、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 3 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

これまでに述べた機能規模測定の考え方にに基づき、提案技法の機能規模測定手順を以下に述べる。

(1) 機能プロセスの識別

4.2 節で述べたように、対話画面を機能プロセスとして識別する。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、その機能部品を基準対話画面に割り当てる。

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	...	データA	データB	...	
入力部品	部品A Eu	Eu	...	Xs	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	Eu Eu		Es			
	X, W	Xu		Xs Xs			
機能規模合計				3		3	0

図 4 機能規模測定テンプレート

(2) データグループの識別

識別された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。なお、1 つのデータグループに対して、同一対話画面中の複数の機能部品が対応する場合がある。COSMIC-FFP 測定法では、これらの機能部品が同一のイベントによって実行される場合は、これらをまとめて 1 つのデータ移動サブプロセスとみなす。

(3) 画面構成部品の分類とデータ移動サブプロセスの割り当て

4.1 節で述べた分類に基づいて、画面構成部品を分類し、機能部品を識別する。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。

(4) 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計数し、対話画面ごとの機能規模を測定する。測定には、図 4 の機能規模測定テンプレートを用いる。まず、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 4 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。各対話画面の機能規模は、式 (1) により算出される。

$$FSS_j = \sum_i DMSP_{ij} \quad (1)$$

ここで、 FSS_j は対話画面 j の機能規模、 $DMSP_{ij}$ 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。 $DMSP_{ij}$ の値はすべて 1 である。機能規模の単位は、COSMIC-FFP 測定法と同様に C_{fsu} を用いる。

測定された各対話画面の機能規模を合計し、被測定ソ



図 5 対話画面の例

ソフトウェアの機能規模合計 FS を式 (2) で算出する。

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

4.4 機能規模の測定例

図 5 に示したカーナビゲーションシステムの対話画面例を対象に、機能規模の測定例を示す。

図 5 の対話画面において、データグループとして「ルート詳細 (IC 名, 距離, 料金, 到着時刻の集合)」が識別できる。入力部品として地図ボタンが識別できる。これは、利用者が指定したルート詳細というデータを受け取り (E_u)、これを他サブシステムに出力する (X_s)。また、出力部品としてルート詳細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 (E_s)、利用者へ出力する (X_u)。戻るボタンは制御部品、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、機能規模は 4 C_{fsu} となる。

5. 画面仕様書に基づく複雑度測定技法

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、 E_u , X_u , E_s , X_s , 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。それぞれ、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 (E_u)

E_u の複雑度は、ユーザからデータを受け取る際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、これらが E_u の複雑度に関する属性となる。これらの属性値の差異に基づいて E_u を分類する。

図 2(a) の E_u 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

部品無し、と識別できる。

(2) ユーザへのエグジット (X_u)

E_u 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が、 X_u の複雑度に関する属性となる。これらの属性値の差異に基づいて X_u を分類する。

図 2(b) の X_u 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無し、と識別できる。

(3) 他サブシステムからのエントリ (E_s)

E_s の複雑度は、他サブシステムからデータを読み込むの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が、 E_s の複雑度に関する属性となる。

(4) 他サブシステムへのエグジット (X_s)

X_s の複雑度は、他サブシステムに対してデータを入力するのに必要な処理量と考えられる。これは、 E_s 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、常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のに必要な処理量と考えられる。したがって、 E_s で示した属性が、 R の複雑度に関する属性となる。

(6) ライト (W)

W の複雑度は、記憶領域に対してデータを入力するのに必要な処理量と考えられる。従って、 X_s で示した属性が、 W の複雑度に関する属性となる。

ここに示したものはデータ移動サブプロセス型の複雑度属性に関する分類である。複雑度の測定時には、データ移動サブプロセスが集約された機能部品について、その分類と複雑度重みに基づいて測定することも可能である。

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与えることで、複雑度という値として測定することができる。

複雑度重みを与える方法として、以下の 3 種類の方法が考えられる。

- (1) 開発工数に基づく重みづけ
- (2) ソースコードに適用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) 困難度の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、ソースコード行数または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に関す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、分類間の困難度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。これは、AHP (Analytical Hierarchy Process) を適用することで容易に行える。

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 複雑度の算出

各データ移動サブプロセス型の分類とそれに対応する複雑度重みが得られると、式(3)によって画面ごとの重みつき機能規模が算出できる。

$$WFSS_j = \sum_i DMSP_{ij} \times W_{ij} \quad (3)$$

ここで、 $WFSS_j$ は対話画面 j の重みつき機能規模、 W_{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表す。

被測定ソフトウェアの重みつき機能規模合計 WFS は、次式で求められる。

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

5.4 データ移動サブプロセス型の分類例

図5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2(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の複数データ属性を出力する。また、ルート詳細情報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付き 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。

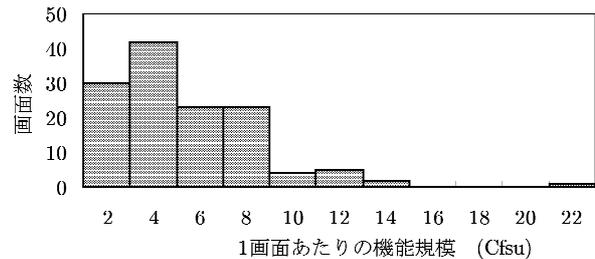


図6 画面あたり機能規模の分布(130画面)

6. 機能規模の測定事例

提案技法を実際のソフトウェア製品に適用し、機能規模の測定を試みた。以下にその結果を述べる。

6.1 対話型ソフトウェア製品の機能規模測定

某社カーナビゲーションソフトウェア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機能規模を測定した。対話部ソフトウェアの対話画面数は全部で170画面、開発に要した工数は約200人月であり、中規模程度あるいは比較的規模の大きいソフトウェア製品といえる。このうち、共通機能を多く含んだ基準画面など14画面を除き、各画面の機能規模を手作業で測定した。

なお、測定対象の画面仕様書からは、各機能部品における Es , Xs , R および W 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必ず出力するものであったため、1つの入力部品に1つの Eu と1つの Xs を割り当てた。同様に、出力部品についても1 Es と1 Xu を割り当てた。したがって、各画面の機能規模は偶数の値となった。

図6に、画面あたり機能規模のヒストグラムを示す。なお、共通機能を基準画面に割り当てて測定した結果、機能規模が測定されなかった画面が多く存在したため、画面数は130画面に減少している。これらの画面の平均機能規模は5.4 Cfsu、機能規模合計は700 Cfsuであった。ヒストグラムからわかる通り、ほとんどの画面は8 Cfsu以下と測定された。

6.2 COSMIC-FFP 実地試験データとの比較

Abranらは、通信、航空制御、防衛、銀行ソフトウェアなど12件のソフトウェアプロジェクトに対して、COSMIC-FFP測定法を試験的に適用した結果を報告している⁸⁾。これらのうち、仕様化、構築およびテスト工程の開発工数データがすべて揃った6件のデータに着目すると、これらの機能規模の範囲は68から8,251 Cfsu、工数の範囲は6から1,300人月で

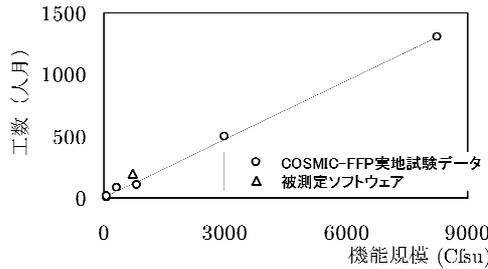


図7 COSMIC-FFP 実地試験データの散布図と回帰直線 (文献8) のデータに基づいて作成)

あった。これらの散布図とデータから得られた回帰直線を図7に示す。データ数は6件と非常に少ないが、相関係数は0.999という極めて高い値が得られた。

これらのデータから、機能規模 FS を説明変数とした開発工数 $Effort$ (人月) の予測式 (5) を得た。

$$Effort = 0.158 \times FS + 2.89 \quad (5)$$

本稿で測定対象とした対話型ソフトウェア製品と文献8)のデータとでは、製品領域が異なっていたり、プログラミング言語等の様々な条件が不明であるため、6.1節の測定結果を式(5)に適用することは適切とはいえない。しかし、ここでは試行として、被測定ソフトウェアの推定機能規模である700 Cfsuを式(5)に代入し、開発工数の算出を行った。その結果、予測工数は113人月であり、実際にかかった約200人月(機能規模の測定対象から除外した14画面を含んだ工数)に対して43%も小さな値であった。この差が条件の差異あるいは複雑度の未考慮などによって生じている可能性もあり、今後、検証実験を行い確かめていく必要がある。

7. おわりに

本稿では、対話型ソフトウェアのCOSMIC-FFP機能規模および複雑度を、画面仕様書に基づいて測定する手法を提案した。また、本技法を実際の対話型ソフトウェア製品に適用し、機能規模の測定を試み、COSMIC-FFP測定法の実地試験データと比較を行った。本稿で述べた内容は未検証の部分が数多くあるため、提案技法の妥当性について確認することは現時点ではできていない。しかし、提案技法の妥当性および有効性が確認できれば、開発初期における効率的な機能規模測定が期待できる。

今後の課題として、本技法によって測定された機能規模が、開発工数やソースコード行数という実績ソフトウェア規模の予測に有効な尺度となるかを確認する必要がある。さらに、複雑度重みを与えることで、予測制度がどの程度向上するのかを検証する必要がある。

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). (小泉寿男ら訳, 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. Soft. 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* (2001).
- 6) Albrecht, A and Gaffney: Software Function, Source Lines of Code, and Development Effort Estimation: A Software Science Validation, *IEEE Trans. Soft. Eng.*, vol.9, no.6, pp.639-648 (1983).
- 7) Fenton, N. E. and Pfleeger, S. L.: *Software Metrics: A Rigorous and Practical Approach*, second edition, Int'l Thomson Computer Press (1997).
- 8) Abran, A., Symons, C. and Oigny, S.: An Overview of COSMIC-FFP Field Trial Results, *12th European Software Control and Metrics*, 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, <http://www.lrgl.uqam.ca/> (2001).
- 13) Jones, C.: *Applied Software Measurement: Assessing Productivity and Quality*, second edition, McGraw-Hill (1996). (鶴保征城ら監訳, ソフトウェア開発の定量化手法 第2版, 共立出版 (1998)).