

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

柏本隆志^{†1} 楠本真二^{†2} 井上克郎^{†2,†3}
鈴木文音^{†1} 湯浦克彦^{†1} 津田道夫^{†4}

ソフトウェアの機能要件を抽出して機能量を定量的に測定するファンクションポイント法が提案されてきている。これまでに多くのソフトウェア開発組織で用いられ、様々な計測手法が提案されてきている。しかし、標準測定法では、ファンクションポイントを計測するための一般的なルールが述べられているにすぎず、詳細な部分の計測には測定者の判断が必要になる。したがって、同一プロダクトに対してのファンクションポイントの計測であっても、計測する人間によって誤差が生じてしまう。また、実際に応用するためには、過去に開発されたソフトウェアに対するファンクションポイントの計測が必要であり、そのためのコストが現場への導入の妨げとなっている。本論文では、これらの問題点を解消するための第1歩として、イベントトレース図に基づく要求仕様書に対してファンクションポイントを計測する手法を提案し、その手法に基づくファンクションポイント計測ツールを試作した。また、試作した計測ツールを実際の要求仕様書に対して適用し、ファンクションポイント計測の熟練者が計測した値との比較を行った。

Function Points Analysis for Requirements Specification Based on Event Trace Diagrams

TAKASHI KASHIMOTO,^{†1} SHINJI KUSUMOTO,^{†2} KATSURO INOUE,^{†2,†3}
AYANE SUZUKI,^{†1} KATSUHIKO YUURA^{†1} and MICHIO TSUDA^{†4}

Function Points Analysis (FPA) was proposed by Albrecht to help measure the size of a computerized business information system. It is widely used to estimate the size and cost of the actual software development. However, it has been reported that since function point counting involves judgment on the part of the counter, some difference for the same product would be caused even in the same organization. We propose detailed FPA measurement method for the requirements specification analyzed using the requirements analysis system REQUARIO, and develop the function points measurement tool based on the method. Moreover, we evaluate the function points measurement tool by applying it to the actual requirements specification developed in practice. Then, we examine the difference between them and discuss the applicability of our tool.

1. ま え が き

近年、コンピュータシステムへの依存度が高まるにつれて、その主要な構成要素であるソフトウェアはま

ずまず大規模化、複雑化、多様化してきている。さらに、ソフトウェアの需要量も増大し、開発期間の短縮化が求められるようになってきた。このような背景から、高い品質を持ったソフトウェアを効率良く開発するために、明確な開発計画の下で開発プロセスの全工程を系統づけて管理する必要性が高まってきている。

明確な開発計画を立てるうえで、開発中に起こる様々な事象を予測し、あらかじめ必要な手段を講じておくことが重要である。ソフトウェア開発に関して予測の対象となるものに規模、投入する工数、開発期間、開発に使用される技術、品質などがある。なかでも重要なものは開発工数と開発期間である。

通常、開発対象ソフトウェアの規模を見積もり、こ

†1 株式会社日立製作所
Hitachi, Ltd.

†2 大阪大学大学院基礎工学研究科
Graduate School of Engineering Science, Osaka University

†3 奈良先端科学技術大学院大学情報科学研究科
Graduate School of Information Science, Nara Institute of Science and Technology

†4 日立システムアンドサービス株式会社
Hitachi System & Services, Ltd.

れに基づいて開発工数と開発期間を予測する手法がとられている⁶⁾。従来、ソフトウェアの規模を表すのにプログラム行数 (SLOC) が使われていた。最近では、ソフトウェアの機能要件だけを抽出して定量的に計測するファンクションポイント法が世界的に普及しつつある。ファンクションポイント法は、Albrecht によって 1979 年に提案された¹⁾。その後、これをベースに種々のファンクションポイント法が提案されている。

しかし、標準測定法ではファンクションポイントを計測するための一般的なルールが述べられているにすぎず、詳細な部分の計測には測定者の判断が必要になる。結果として、同一プロダクトに対してのファンクションポイントの計測であっても、計測する人間によって誤差が生じてしまうという問題点が指摘されている⁴⁾。たとえば、同じ組織内の人間が同じプロダクトに対して測定した場合は 12%、違う組織の人間が測定した場合は 30% 以上の誤差が出るという報告もされている³⁾。また、ファンクションポイントを実際に応用して見積りを行うためには、そのソフトウェア開発組織で過去に開発されたソフトウェアのファンクションポイント値が基礎データとして必要であり、その計測のためのコストが現場への導入の妨げとなっている。したがって、新規開発のソフトウェアだけでなく、既存ソフトウェアに対するファンクションポイント計測を効率良く行うことが重要な課題となっている。

本論文では、これらの問題点に対処するための第 1 歩として、イベントトレース図⁸⁾に対してファンクションポイントを計測するための指針を提案する。さらに、提案した方法の有効性を評価するために、CASE ツールで作成されたイベントトレース図に基づく要求仕様書に対して、ファンクションポイントを計測するための計測ツールの試作を行った。最後に、実際の要求仕様書に対してツールの計測したファンクションポイント値とファンクションポイント計測の熟練者が測定したファンクションポイント値との比較を行った。その結果、仕様書に記述されている機能については、熟練者が計測した値とツールが計測した値が一致した。

以降、2 章ではファンクションポイント法の概要を紹介する。3 章ではイベントトレース図に対するファンクションポイント計測ルールについて提案する。4 章では、提案したルールを評価するためのケーススタディについて述べ、最後に 5 章ではまとめと今後の課題について述べる。

2. ファンクションポイント法

2.1 概要

ファンクションポイント法は、Albrecht によって 1979 年に提案された¹⁾。その後、これをベースに様々な改良手法が提案されており、現在では、数十種類の計測方法がある。たとえば、IBM 法⁶⁾、IFPUG 法²⁾、3D Function Points 法⁵⁾、Feature Points 法⁵⁾、MarkII 法¹⁰⁾ などである。現在では、IFPUG (International Function Point Users Group) 法²⁾ が主流技法となっている。IFPUG 法は、Albrecht 版の使い勝手や曖昧な部分を改良したバージョンであり、欧米で広く使用されている。

ファンクションポイント法は、利用者要求のうちの機能要求仕様の大きさを定量的に計測する手法であり、求められる計測値は、機能量または機能規模と呼ばれる。また、機能量の単位としては、伝統的にファンクションポイントという呼称が使われている。

機能量の計測では、計測対象ソフトウェアの機能のうち、画面や帳票、ファイルなどを通じた情報の入出力に着目し、それらを種類別に数え上げ、種類数を加重合計した値を機能量とする。ただし、実現方法の違いによる影響を除くために、画面などの数は物理設計結果ではなく論理設計レベルで数える。

こうして得られた機能量の規模尺度としての最大の長所は、

- 規則に従って計測される値 (推定値でなく誰が計測しても同じ値が得られる)
- 機能仕様だけに依存 (開発環境や開発言語などの技術要件に左右されない)

という 2 点である²⁾。

これに対して、従来からソフトウェアの規模尺度として使われてきた SLOC 尺度は、開発言語や開発環境が変わると値が変化し、プログラムが完成するまでは値が確定しない、などの点で機能量尺度より劣っている。

本論文では、数多くのファンクションポイント法のなかから、主流技法となっている IFPUG 法を基にしてファンクションポイントを計測する。

2.2 IFPUG 法

IFPUG 法は、Albrecht 版のファンクションポイント法に対して複雑さの評価の客観化やルールの精密化・適正化などの変更を行ったバージョンである。

IFPUG 法は次の Step1 ~ Step7 でファンクションポイントを計測する²⁾。

Step1 (算出種類の選択): 算出種類を、アプリケー

ション FP (アプリケーションソフトウェアの大きさを表すファンクションポイント), 新規開発プロジェクト FP (新規にアプリケーションを開発するプロジェクトの規模を知るために使用するファンクションポイント), 機能改良プロジェクト FP (すでに存在するアプリケーションを拡張するプロジェクトの規模を知るために使用するファンクションポイント) の 3 種類のなかから選択する .

Step2 (計測境界の設定): ファンクションポイントを計測する対象 (範囲) を明確にするため , 計測境界の設定する . 通常は 1 つのアプリケーションを範囲にするが , そのアプリケーションが非常に大きい場合は , 計測単位をサブシステムなどの機能単位に分割する .

Step3 (データファンクションの計測): データファンクションとは , アプリケーション中にあり , ユーザが認識できる論理的な意味でのデータのまとまりのことである . アプリケーションのなかからデータファンクションを抽出し , ファンクションタイプを決定し , 複雑さを測定する .

データファンクションには以下の 2 種類のファンクションタイプがある .

- 内部論理ファイル (Internal Logical File): 計測対象のアプリケーション内でデータが更新される論理的な関連を持ったデータの集合 .
- 外部インタフェースファイル (External Interface File): 計測対象のアプリケーションによってデータが参照されるデータの集合 (データは更新されない) .

次に , それぞれのデータファンクションをレコード種類数 , データ項目数という 2 つのパラメータによって , 低・中・高の 3 段階に重み付けする .

Step4 (トランザクションファンクションの計測): アプリケーションに対するデータの出入りをともなう処理をトランザクションファンクションという . アプリケーションのなかからトランザクションファンクションを抽出し , ファンクションタイプを与え , 複雑さを測定する .

トランザクションファンクションには以下の 3 種類のファンクションタイプがある .

- 外部入力 (External Input): 計測境界外からのデータ入力によってデータファンクションの更新を行う処理 .
- 外部出力 (External Output): (オンライン処理における) 計測境界外へのデータ出力を含む処理のうち , 出力データに派生データ (計

算や条件判断など何らかの加工を必要とするデータ項目) を含むもの .

- 外部照会 (External Inquiry): (オンライン処理における) 計測境界外へのデータ出力を含む処理であるもので , 出力データに派生データを含まないもの . また , 処理が内部論理ファイル (ILF) を更新しないもの .

次に , それぞれのトランザクションファンクションを関連ファイル数 , データ項目数という 2 つのパラメータによって , 低・中・高の 3 段階に重み付けする .

Step5 (未調整ファンクションポイント算出): Step3 , Step4 の結果を基に , 種類別重み別に個数を数え , 表 1 を用いて算出した結果が複雑さの評価値を加え合わせる . この合計値が未調整ファンクションポイント (Unadjusted Function Points) となる .

Step6 (調整係数の算出): 未調整ファンクションポイントは「データのまとまり」と「データの出入り」のみに着目した値で , 性能 , 信頼性 , ユーザインタフェースなどは考慮していない . そこでシステム特性をファンクションポイントに反映させるために , 表 2 に示すシステム特性の 14 項目を 6 段階で評価し , その結果から調整係数を算出

表 1 未調整ファンクションポイント算出表

Table 1 Table for unadjusted function point counting.

	低	中	高	合計
ILF	$\square \times 7 = \square$	$\square \times 10 = \square$	$\square \times 15 = \square$	
EIF	$\square \times 5 = \square$	$\square \times 7 = \square$	$\square \times 10 = \square$	
EI	$\square \times 3 = \square$	$\square \times 4 = \square$	$\square \times 6 = \square$	
EO	$\square \times 4 = \square$	$\square \times 5 = \square$	$\square \times 7 = \square$	
EQ	$\square \times 3 = \square$	$\square \times 4 = \square$	$\square \times 6 = \square$	
			未調整 FP	

表 2 システム特性の 14 項目

Table 2 Adjustment factors.

1	データ通信機能
2	分散データ処理
3	性能条件
4	高負荷構成
5	トランザクション率
6	オンラインデータ入力
7	エンドユーザの効率
8	オンライン更新
9	複雑な処理
10	再利用性
11	インストレーションの容易さ
12	運用の容易さ
13	複数サイト
14	変更の容易さ

調整係数 = $0.01 \times$ 影響度の合計 + 0.65

する。

調整係数は、ファンクションポイント算出の際に未調整ファンクションポイントを補正する役割がある。

Step7 (最終ファンクションポイント算出): 未調整ファンクションポイントと調整係数を用いて、最終ファンクションポイントを算出する。

2.3 ファンクションポイント計測ツール

これまでに多くのファンクションポイント計測ツールが開発され、市販されている。

たとえば、構造化設計を支援するための CASE ツールである SAVER には、ファンクションポイント計測機能が実装されている¹³⁾。これはデータフロー図上のプロセスをトランザクションファンクションに、ファイルをデータファンクションに対応させてファンクションポイントを計測している。しかし、ファンクションのタイプや複雑度をユーザがすべて指定しなければならない。同様に、ABT Corporation の Function Point Manager も IFPUG 法に基づいて、ファンクションポイントを計測してくれるが、計算の部分が自動化されているだけであり、トランザクションファンクション、データファンクション、複雑度などの基本データはすべてユーザが入力する必要がある。

オラクル社の Designer/2000 や VIASOFT の Visual Recap¹¹⁾ にも、ファンクションポイント計測機能が存在する。これは自動的に計算を行うが、たとえば、データファンクションにおける内部論理ファイルと外部インタフェースファイルの区別をせず、すべて内部論理ファイルと計測してしまう。また、通常データファンクションと見なさない一時ファイルも内部論理ファイルと計測してしまうため、実際の値に比べて数十倍数十倍以上の差が出る可能性がある。

3. イベントトレース図に対するファンクションポイント計測

3.1 イベントトレース図

通常、情報システム構築の上流工程では、システム分析者はユーザに、文章、表、図などで作成した仕様書を提示し、ユーザはその仕様書で内容を確認している。その際、システムの静的な情報については、文章や図として表現することは可能であるが、動的な情報を記述することは難しい。

動的な情報を表現するための手段として、イベント

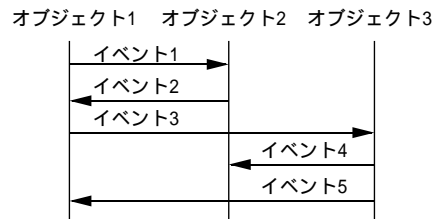


図 1 イベントトレース図の例

Fig. 1 Example of event trace diagram.

トレース図⁸⁾(シーケンス図)が広く用いられている。イベントトレース図は、システムに存在するオブジェクト間のメッセージ送受信によって発生する事象の推移を定義した図である。図 1 にイベントトレース図の例を示す。矢印を用いてオブジェクト間のメッセージ送受信の様子を定義する。横軸がオブジェクトの並びで、縦軸が時間である。

3.2 ファンクション識別方法

2章で述べたように、ファンクションポイント計測に必要な情報は、計測境界を通じて、ユーザ、あるいは、外部アプリケーションが計測対象アプリケーションとの間で行う処理と扱われるデータの情報である。イベントトレース図では、オブジェクトとして計測対象アプリケーションシステム内のデータベースやコンポーネント、ユーザや外部アプリケーションが記述される。また、実行される処理もメッセージ送受信という形で記述される。したがって、データファンクションの候補にイベントトレース図上の人間以外のオブジェクトを、トランザクションファンクションの候補にメッセージのやりとりを行っている 2 つのオブジェクト対を、それぞれ選定すればよい。

次に、データファンクションとトランザクションファンクションのタイプ分けについて考える。トランザクションファンクションについては、メッセージのやりとりの前後で送信側のオブジェクトが受信側のオブジェクトに送ったデータの内容と受信側のオブジェクトの持つデータが変更されたかどうかを調べることで、外部入力、外部出力、外部照会の場合分けが可能である。具体的には、送信側オブジェクト A から受信側オブジェクト B へメッセージが送られる場合、オブジェクト A がデータファンクションである場合とない場合で以下の 3 通りに分類し、R1~R8 のルールを用いてタイプ分けを行う。

- (1) オブジェクト A がデータファンクションの候補でなく、オブジェクト B がデータファンクションの候補の場合

R1: B のデータが変更されていれば外部入力。

ここで述べたもの以外にも、コンサルティング会社、各種メーカーでは自社の計測ツールを所有しているといわれている。しかし、情報が公開されていないためその詳細は不明である。

R2: B のデータが変更されていない場合は外部照会。

(2) オブジェクト A がデータファンクションの候補で、オブジェクト B がデータファンクションの候補でない場合

R3: A の出力データが派生データであれば外部出力。

R4: A の出力データが派生データでなければ外部照会。

(3) オブジェクト A と B の両方がデータファンクションの候補の場合

R5: B のデータが変更され、A の出力データが派生データであれば、外部入力と外部出力。

R6: B のデータが変更され、A の出力データが派生データでなければ、外部入力と外部照会。

R7: B のデータが変更されず、A の出力データが派生データであれば、外部照会と外部出力。

R8: B のデータが変更されず、A の出力データが派生データでない場合、1 つの外部照会。

ただし、外部照会と判定されたオブジェクト対については、それと対になるオブジェクト対を調べ、複雑度の比較を行い、複雑度の高い方のみを外部照会と見なす必要がある (R8 ではその処理を同時に行う)。

一方、データファンクションについては、すべてのトランザクションファンクションの場合分けを行った後で、データファンクション候補のオブジェクトのなかで、データが一度でも変更されているものを内部論理ファイル、一度も変更されていないものを外部インタフェースファイルと判定すればよい。

4. ケーススタディ

3.2 節で述べた手法の適用可能性を確認するために、ある CASE ツールで作成されたイベントトレース図に基づく要求仕様書に対して実装を行う。具体的には、我々の一部が開発に携わった要求支援システム REQUARIO⁹⁾ で作成されたイベントトレース図 (シナリオ図) を対象とする。

4.1 対象となる CASE ツール

REQUARIO⁹⁾ は開発するシステムへの要求仕様をアニメーションで表示するツールである。

REQUARIO では、ストーリー、シナリオ、シーン、キャラクターという概念で要求仕様を定義する。これらの概念構成を図 2 に示す。このなかのシナリオがイベントトレース図に対応し、さらにキャラクターがオブジェクトに対応する。

(1) ストーリー: 要求仕様の断片的な具体例の集合

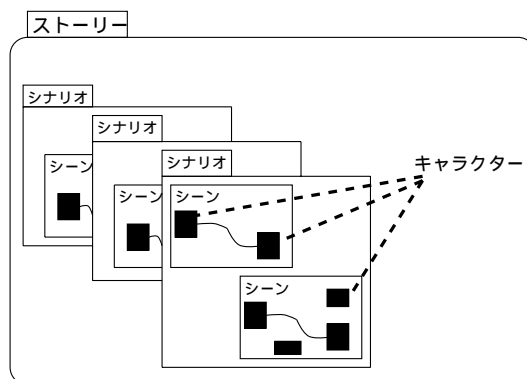


図 2 REQUARIO の構成
Fig. 2 Overview of REQUARIO.

である。REQUARIO の成果物をストーリーという単位で管理している。

(2) シナリオ: ストーリー中の「要求仕様の断片的な具体例」をシナリオという。シナリオは、ストーリーに登場する「ひと」あるいは「もの」と、その「ふるまい」(動作・処理)の流れから構成されている。つまり、「誰が(どこが)」、「誰に(どこに)」、「何を(どのようなイベントを)」発行するかを表したものである。

(3) シーン: ストーリーに登場する「ひと」あるいは「もの」をキャラクタといい、その「ふるまい」(動作・処理)の流れをシーンという。REQUARIO は、最小単位であるシーンを定義していくことでシナリオを作っていく。シーンには、次のような表現を付加することができる。

- キャラクタが持っているデータが自分自身または他のキャラクタのデータ代入によって変化する様子。
- ふるまいに関係するキャラクタがシーンに関与する様子 (関与線と呼ぶ)。
- メッセージ呼び出しが発生する条件 (起動条件と呼ぶ)。起動条件を設定することで、実際の要求仕様のなかに出てくる「ある条件のときに、こういう処理を行う」というようなシナリオを表現することが可能となる。

(4) キャラクタ: ストーリーに登場する「ひと」あるいは「もの」をキャラクタという。REQUARIO ではシステムに登場する人、データの集まりなどの動作の主体を“人”、“データベース”などのアイコンで表示する。キャラクタ同士がシーンを構成していくことでシナリオとなり、ストーリーとなる。キャラクタは 2 種類の情報を持っている。1 つは、キャラクタの持つデータを表現した“属

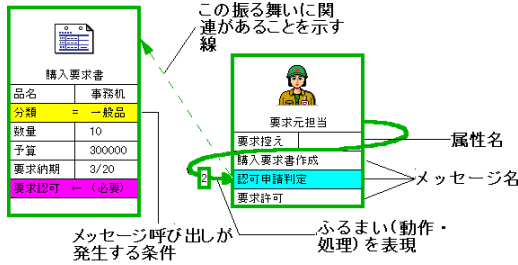


図 3 シーン例(1)

Fig. 3 Example of scene (1).

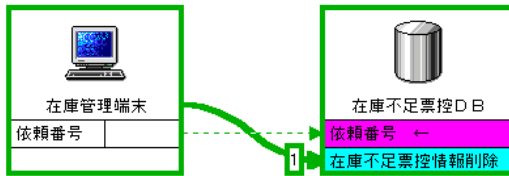


図 4 シーン例(2)

Fig. 4 Example of scene (2).

性”と、もう1つは、キャラクタの持つふるまいを表現した“メッセージ”である。

REQUARIOでの記述例を2つ示す。図3は要求元担当が購入要求書を見て、購入要求書の分類という項目が一般品であれば購入を認可するという処理を表す。このシーンでは購入要求書がシーンに關係するキャラクタとして登場する。そして、購入要求書はデータ入力が行われ更新されていることを表現している。

図4は在庫管理端末から在庫不足票控DBに対してメッセージが送られているシーンである。このシーンでは在庫管理端末からデータが出力され、在庫不足票控DBへデータが入力されている。そして、このデータ入力によって在庫不足票控DBのデータが更新されている。

4.2 ファンクションポイント計測ツール

4.2.1 計測概要

ここではツールで実装するファンクションポイント計測の概要について説明する。4.1節で述べたように、REQUARIOの記述はイベントトレース図よりも多くの情報を含んでいるので、Step3とStep4のデータファンクションとトランザクションファンクションの抽出については、3章で述べたルールを拡張する必要がある。

Step1 (算出種類の選択): 算出種類はアプリケーションファンクションポイントに限定する。

Step2 (計測境界の設定): 算出種類をアプリケーションファンクションポイントに限定しているた

め、Step1の入力として与えられた仕様書上で、ユーザとアプリケーションの境界が計測境界となる。

Step3 (データファンクションの計測): 指定したストーリーファイル中のデータファンクションを、4.2.2項で説明するルールに従って抽出する。

Step4 (トランザクションファンクションの計測): 指定したストーリーファイル中のトランザクションファンクションを、4.2.3項で説明するルールに従って抽出する。

Step5 (未調整ファンクションポイント算出): Step3, Step4の結果を基にファンクションタイプ別複雑さ別に個数をカウントし、それぞれに対して重み付けをし、加え合わせて未調整ファンクションポイントを算出する。

Step6 (調整係数の算出): システム特性の影響度をユーザが入力し、調整係数を算出する。

Step7 (最終ファンクションポイント算出): 未調整ファンクションポイントと調整係数から最終ファンクションポイントを算出する。

4.2.2 データファンクション抽出ルールの調整

3.2節で述べたルールに基づいてデータファンクションの抽出は以下のように行う。

- (1) データベースのアイコンで表示されているキャラクタをデータファンクションとする。
- (2) (1)で抽出されたデータファンクションのなかで、データを更新されるものを内部論理ファイル(ILF)、更新されないものを外部インタフェースファイル(EIF)とする。
- (3) レコード種類数(RET)は1種類、データ項目数(DET)はキャラクタの属性の数とし、複雑さ決定表を用いて各データファンクションの複雑さを判定する。

なお、ツールの実装としては、ユーザは計測対象のシステムに出現するすべてのキャラクタに対して、データファンクションであるかないかの指定を行うこともできるようにする。

4.2.3 トランザクションファンクション抽出ルールの調整

3.2節で述べたルールに基づいてトランザクションファンクション抽出ルールを図5のように拡張する。ルール3とルール4を、イベントトレース図に關与線

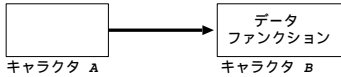
REQUARIOでは複数のデータ項目をサブセットして定義することができないため1となる。ただし、レコード種類数の値が確定している場合には、4.2.5項で述べるツールの機能を用いて設定することで計測は可能である。

という情報が付加されたことにより追加した。

外部照会と外部出力の決定は、出力データが派生データであるかどうかで行う。派生データの判別は、データ出力をしたキャラクタの属性とそのデータを受け取ったキャラクタの属性を比較し、異なる属性を持っていれば派生データと考えられる。

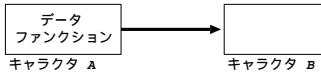
抽出されるトランザクションファンクションが外部照会の場合は、それと対になる外部照会がないかを調べる。対となるシーンがあれば複雑度の比較を行い、複雑度の高い方を外部照会にする。各トランザクションファンクションの複雑さについては、関連ファイル

ルール1



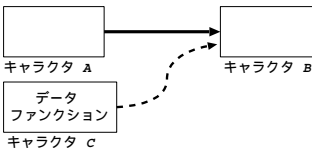
- ルール1.1 キャラクタBのデータが変更されていれば外部入力。
 ルール1.2 キャラクタBのデータが変更されていなければ外部照会。

ルール2



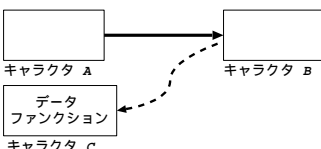
- ルール2.1 キャラクタAの出力データが派生データであれば外部出力。
 ルール2.2 キャラクタAの出力データが派生データでなければ外部照会。

ルール3



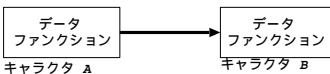
- ルール3.1 キャラクタCの出力データが派生データであれば外部出力。
 ルール3.2 キャラクタCの出力データが派生データでなければ外部照会。

ルール4



- ルール4.1 キャラクタCのデータが変更されていれば外部入力。
 ルール4.2 キャラクタCのデータが変更されていなければ外部照会。

ルール5



- ルール5.1 キャラクタBのデータが変更され、キャラクタAの出力データが派生データのとき、外部入力と外部出力。
 ルール5.2 キャラクタBのデータが変更され、キャラクタAの出力データが派生データでないとき、外部入力と外部照会。
 ルール5.3 キャラクタBのデータが変更されず、キャラクタAの出力データが派生データのとき、外部照会と外部出力。
 ルール5.4 キャラクタBのデータが変更されず、キャラクタAの出力データも派生データでないとき、外部照会。

図5 トランザクションファンクション判別ルール

Fig. 5 Rules for transactional function counting.

数(対象となるトランザクションファンクションの処理中にデータが更新または参照されるデータファンクションの個数)とデータ項目数を数えあげることによって判定する。

さらに、シーンに起動条件が存在するとき、その起動条件を含んでいるキャラクタがデータファンクションならば外部照会の機能を追加する。起動条件の判定には、なんらかの照会処理を行う必要があるからである。たとえば、図3で購入要求書がデータファンクションである場合を考える。まず、ルール4から外部入力抽出できる。また、“分類=一般品”という起動条件から、購入品が一般品であるかどうかを判定するための照会処理が必要となるので、外部照会も抽出する。

まとめると、計測ツールにおけるトランザクションファンクションの抽出手順は以下のようになる。

- (1) ルール(図5参照)に従い、トランザクションファンクションを抽出し、その種類(外部入力、外部出力、外部照会)を決定する。
- (2) 関連ファイル数(FTR)はシーンに関連しているデータファンクションの数、データ項目数(DET)は入出力されたデータファンクションの属性の数とし、複雑さ決定表を用いて、各トランザクションファンクションの複雑さを判定する。

4.2.4 ファンクションポイント計測ツールの概要

Windows95/98上で動作するツールをMFC(クラスライブラリ)を用いて、C++で実装した。プログラムサイズは約7000行である。ツールの入力はREQUARIOで作成された要求仕様書で、ファンクションポイントの計測結果とその計測のもとになる各ファンクションの候補を出力する。

システム構成を図6に示す。

- 解析部：REQUARIOの出力ファイルを構文解析し、ファンクションポイント計測のために必要であるデータを抽出する。
- 解析DB：解析部で抽出したデータを格納しておく。
- 計測部：ファンクションポイントを計算する。
- 計測DB：計測結果(ファンクションポイント、データファンクション、トランザクションファンクション)を格納しておく。
- インタフェース部：計測結果(ファンクションポイント、データファンクションの候補、トランザクションファンクションの候補)を表示する。ツールの画面例を図7に示す。

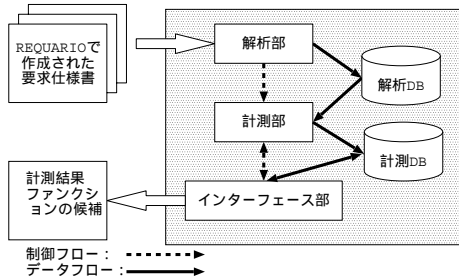


図6 システム構成
Fig.6 System architecture.

図7の(a)はメインスクリーンである．この画面では各ファンクションの抽出された個数，調整係数の値，未調整・最終ファンクションポイントの値を表示する．

図7の(b)はキャラクタの一覧表示画面である．この画面では，要求仕様書に登場したキャラクタの情報(キャラクタ名，ビットマップの種類，ビットマップ名，レコード種類数，データ項目数，ファンクションタイプ，ファンクションの複雑度)を表示する．また，データファンクションとして抽出されたキャラクタも表示する．

図7の(c)はシーンの一覧表示画面である．この画面では，要求仕様書に登場したシーンの情報(シナリオ名，送信キャラクタ，受信キャラクタ，関与キャラクタ，メッセージ名，関連ファイル数，データ項目数，ファンクションタイプ，ファンクションの複雑度)を表示する．また，トランザクションファンクションとして抽出されたシーンも表示する．

4.2.5 結果の調整

ツールで自動計測した後，各ファンクションの候補を確認でき，このときにファンクションとして抽出できていないものやファンクションの種類が間違っているものが存在すれば，キャラクタ一覧表示画面やシーン一覧表示画面から変更し，ファンクションポイントの再計算を実行できる．

この調整機能を用いて，自動的に判断できない部分の判断を計測者が行うことにより，より正確にファンクションポイントを求めることが可能となる．

4.3 適用対象

ツールの計測したファンクションポイント値が妥当なものであるかを評価するために計測ツールを実際の要求仕様書に対して適用した．具体的には，REQUARIOで作成された購入業務，発注業務，在庫管理の要求仕様書の評価対象とし，ツールの計測したファンクションポイント値とファンクションポイント計測の熟練者が測定したファンクションポイント値との比

ファンクションタイプ	複雑度			合計
	低	中	高	
内部処理ファイル	1 × 7 = 7	0 × 10 = 0	0 × 15 = 0	7
外部インターフェースファイル	1 × 5 = 5	0 × 7 = 0	0 × 10 = 0	5
外部入力	2 × 3 = 6	0 × 4 = 0	0 × 6 = 0	6
外部出力	0 × 4 = 0	0 × 5 = 0	0 × 7 = 0	0
外部照会	8 × 3 = 24	0 × 4 = 0	0 × 6 = 0	24
未調整ファンクションポイント				42
影響度の合計				35
最終ファンクションポイント				42.00

(a) メインスクリーン

キャラクタ名	ビットマップの種類	ビットマップ名	レコード種類数	データ項目数	ファンクションタイプ	複雑度
在庫不足照会DB	その他	ドラム(DB)	1	4	内部処理ファイル	低
積荷DB	その他	ドラム(DB)	1	6	外部インターフェースファイル	低
積荷票	帳票	伝票(帳票)	0	6	なし	なし
在庫不足照会	帳票	伝票(帳票)	0	7	なし	なし
出荷指示書	帳票	伝票(帳票)	0	4	なし	なし
在庫不足照会	帳票	伝票(帳票)	0	3	なし	なし
客寄せ	人間	安全ヘルプシート	0	3	なし	なし
配送係	人間	作業服(納入)	0	3	なし	なし
発行係	人間	オペレーション	0	6	なし	なし

(b) キャラクタの一覧表示

シーン名	シナリオ名	送信キャラクタ	受信キャラクタ	関与キャラクタ	メッセージ名	データ項目数	ファンクションタイプ	複雑度
購入	受注	なし	なし	なし	在庫更新	0	外部照会	低
発注	受注	積荷票	なし	なし	積荷更新	0	外部照会	低
在庫管理	受注	積荷票	なし	なし	積荷更新	0	外部照会	低
在庫不足照会	受注	なし	なし	なし	在庫不足照会	0	外部照会	低
出荷指示	受注	なし	なし	なし	出荷指示作成	0	外部照会	低
客寄せ	受注	なし	なし	なし	客寄せ	0	外部照会	低
配送	受注	なし	なし	なし	配送	0	外部照会	低
発行	受注	なし	なし	なし	発行	0	外部照会	低
購入	受注	なし	なし	なし	在庫更新	1	外部照会	低
発注	受注	なし	なし	なし	積荷更新	1	外部照会	低
在庫管理	受注	なし	なし	なし	積荷更新	1	外部照会	低
在庫不足照会	受注	なし	なし	なし	在庫不足照会	1	外部照会	低
出荷指示	受注	なし	なし	なし	出荷指示作成	2	外部照会	低
客寄せ	受注	なし	なし	なし	客寄せ	0	外部照会	低
配送	受注	なし	なし	なし	配送	0	外部照会	低
発行	受注	なし	なし	なし	発行	0	外部照会	低
購入	受注	なし	なし	なし	在庫更新	0	外部照会	低
発注	受注	なし	なし	なし	積荷更新	0	外部照会	低
在庫管理	受注	なし	なし	なし	積荷更新	0	外部照会	低
在庫不足照会	受注	なし	なし	なし	在庫不足照会	0	外部照会	低
出荷指示	受注	なし	なし	なし	出荷指示作成	0	外部照会	低
客寄せ	受注	なし	なし	なし	客寄せ	0	外部照会	低
配送	受注	なし	なし	なし	配送	0	外部照会	低
発行	受注	なし	なし	なし	発行	0	外部照会	低
購入	受注	なし	なし	なし	在庫更新	1	外部照会	低
発注	受注	なし	なし	なし	積荷更新	1	外部照会	低
在庫管理	受注	なし	なし	なし	積荷更新	1	外部照会	低
在庫不足照会	受注	なし	なし	なし	在庫不足照会	1	外部照会	低
出荷指示	受注	なし	なし	なし	出荷指示作成	1	外部照会	低
客寄せ	受注	なし	なし	なし	客寄せ	0	外部照会	低
配送	受注	なし	なし	なし	配送	0	外部照会	低
発行	受注	なし	なし	なし	発行	0	外部照会	低

(c) シーンの一覧表示

図7 ファンクションポイント計測ツールの画面例
Fig.7 Output from FP measurement tool.

較を行った．
購入業務の要求仕様書には，備品を購入するときには様々な書類が作成されるが，そのときの業務の流れを記述してある．発注業務の要求仕様書には，備品を発注する際に行われる購入金額の見積りや発注先を決定する業務の流れが記述してある．在庫管理の要求仕様書には，商品を倉庫に管理したり，商品の生産計画を立てたりする業務の流れが記述してある．
これらのデータのシナリオ数，シーン数，キャラクタ数を表3に示す．なお，ツールを用いてファンクションポイントを計測する際には，データファンクションを指定しなければならないが，今回は，熟練者がデータファンクションと判別したキャラクタを指定した．またツールのファンクションポイント値としては，計測結果の調整をしていない値を用いることにした．

表 3 通用データ
Table 3 Summary of data.

	シナリオ数	シーン数	キャラクタ数
購入業務	7	15	6
発注業務	9	37	14
在庫管理	2	11	15

表 4 適用結果
Table 4 Results.

	熟練者の FP 値			ツールの FP 値		
	購入	発注	在庫	購入	発注	在庫
DF の FP 値	14	29	24	14	29	24
TF の FP 値	18	63	36	15	44	27
FP 値の合計	32	92	60	29	73	51

4.4 適用結果と考察

3つのシステムの要求仕様書に対してツールの計測したファンクションポイント値と熟練者が計測したファンクションポイント値を表4に示す。比較の結果、ツールが計測した値が熟練者の測定した値に対して、約80%の精度であった。

表4と各ファンクションの計測結果から、ツールでは抽出できていないトランザクションファンクションがあるという結果が得られた。ツールと熟練者の計測結果からこの原因について分析を行った。

熟練者がファンクションポイントを計測する際に作成した資料とツールの出力結果を照合したところ、差の発生した原因は、熟練者がREQUARIOの記述にシステム化する際に必要と思われる処理を追加して計測しているためであることが確認できた。差が発生した部分について、熟練者が追加した処理をREQUARIOの記述に追加してツールで計測を行った結果、同じ値が計測されることが確認できた。

この結果は、計測ツールを用いてファンクションポイントを計測するときには、入力となる要求仕様書の詳細度のある保証する必要があることを示唆している。計測ツールに入力する前に、入力となる要求仕様書に対してレビューを実施して、必要な機能が正しく反映されているということを確認することが実用上必要である。

5. まとめと今後の課題

イベントトレース図に対してファンクションポイントを計測するための指針を示し、その指針に基づいてあるCASEツールで作成された要求仕様書に対して、ファンクションポイントを求めるための計測ツールを試作した。さらに開発したツールを実際の開発現場で適用し、ツールが計算した値とファンクションポイン

ト計測の熟練者が測定した値を比較を行った。

今後は、試作したツールについては、数多くの要求仕様書に対する適用とルールの変更が考えられる。今回の適用ではツールの計測結果と熟練者の計測結果には大きな差は見られなかったが、今後様々な要求仕様書からファンクションポイントを求めて計測ルールを改善していく必要がある。また、今回の実装ではIFPUG CPM4.0に基づいているが、新しくリリースされたCPM4.1に対応させることも必要である。

さらに、提案した指針は、イベントトレース図に基づく仕様書全般に適用可能である。たとえば、オブジェクト指向設計で作成された設計仕様書を対象として、ファンクションポイントを計測するツールを開発することも重要な課題である。現在、Rational Rose⁷⁾を用いて記述された設計書に対してファンクションポイントを求めるツールの開発を行っている¹²⁾。

参考文献

- 1) Albrecht, A.J.: Measuring Application Development Productivity, *Proc. Joint SHARE, GUIDE, and IBM Application Development Symposium*, pp.83-92 (1979).
- 2) IFPUG: Function Point Counting Practices Manual, Release 4.0, International Function Point Users Group (1994).
- 3) Kitchenham, B.A.: The Problem with Function Points, *IEEE Software*, Vol.14, No.2, pp.29-31 (1997).
- 4) Low, G.C. and Jeffery, D.R.: Function Points in the Estimation and Evaluation of the Software Process, *IEEE Trans. Softw. Eng.*, Vol.16, No.1, pp.64-71 (1990).
- 5) 中村 永: 科学技術計算とリアルタイム制御に向くソフト計測手法, 日経エレクトロニクス, No.658, pp.175-185 (1996).
- 6) 西山 茂: ソフトウェア規模の見積り技術の最近の流れ—行数による評価から機能量による評価へ, 情報処理, Vol.35, No.4, pp.289-298 (1994).
- 7) Rational Software Corporation: UML Notation Guide, version 1.1 (1997).
- 8) Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F. and Lorenzen, W.: *Object Oriented Modeling and Design*, Prentice Hall (1991).
- 9) 斎藤, 大成, 湯浦, 亀田: オブジェクト指向による要求仕様書視覚化ツール“REQUARIO”, 日立評論, Vol.77, No.12, pp.15-18 (1995).
- 10) Symons, C.: *Software Sizing and Estimating*, John Wiley & Sons (1991).
- 11) Visual Recap: *Quality Measurement*, VIA-SOFT (1997).
- 12) 上村, 柏本, 橋本, 井上: UMLで記述された設

計仕様書からのファンクションポイント計測手法，1999年電子情報通信学会総合大会情報・システム講演論文集1，D-3-9 (1999).

- 13) SAVER を利用したファンクションポイント法の実現，ヴェストソフトウェア (1997).

(平成 11 年 7 月 12 日受付)

(平成 12 年 4 月 6 日採録)



柏本 隆志 (正会員)

平成 9 年大阪大学基礎工学部情報工学科卒業．平成 11 年同大学院博士前期課程修了．同年 (株) 日立製作所入社，現在に至る．ソフトウェア工学の研究に従事．



楠本 真二 (正会員)

昭和 63 年大阪大学基礎工学部情報工学科卒業．平成 3 年同大学院博士課程中退．同年同大学基礎工学部情報工学科助手．平成 8 年同大学講師．平成 11 年同大学助教授．工学博士．ソフトウェアの生産性や品質の定量的評価，プロジェクト管理に関する研究に従事．電子情報通信学会，IEEE 各会員．



井上 克郎 (正会員)

昭和 54 年大阪大学基礎工学部情報工学科卒業．昭和 59 年同大学院博士課程修了．同年同大学基礎工学部情報工学科助手．昭和 59～61 年ハワイ大学マノア校情報工学科助教授．平成元年大阪大学基礎工学部情報工学科講師．平成 3 年同学科助教授．平成 7 年同学科教授．工学博士．ソフトウェア工学の研究に従事．電子情報通信学会，日本ソフトウェア科学会，IEEE，ACM 各会員．



鈴木 文音 (正会員)

1994 年東京工業大学電気電子工学修士課程修了．同年 (株) 日立製作所に入社．ソフトウェアの生産性やプロジェクト管理に関する開発研究に従事．



湯浦 克彦 (正会員)

昭和 53 年東京大学工学部精密工学科卒業．昭和 55 年同大学院修士課程修了．同年 (株) 日立製作所入社，現在ビジネスソリューション開発本部に所属．自然言語処理，プログラミング言語およびオブジェクト指向設計に関する研究開発およびオブジェクト指向型の大規模情報システムの開発に従事．ソフトウェア科学会会員．



津田 道夫 (正会員)

昭和 45 年同志社大学工学部電気工学科卒業．同年 (株) 日立製作所入社．平成 11 年 (株) 日立システムアンドサービス産業・流通システムサービス事業部所属．現在に至る．ソフトウェア生産技術の企画・開発に従事．