

オブジェクトの進化プロセスにおける統計的観測

中谷 多哉子[†] 玉井 哲雄^{††}

1 クラスあたりの行数やメソッド数, 1 メソッドあたりの行数といった計測値の頻度分布は, 左に最頻値を持ち右に尾を引く形状を持つ。この分布の形状は開発言語やシステムの進化に依存しない形状として知られているが, この分布には, 漸進的开发によって, 尾が長くなったり, 最頻値の頻度が増加して山が高くなるといった変化が起こることもある。我々は, 計測値の分布が開発過程によって変化する点に着目し, 分布の変化と開発過程の因果関係を調査することにした。

本研究では, オブジェクトの進化プロセスを定量的に観測した計測値の頻度分布に対して, 負の二項分布の適用を試み, 適合度検定を行った。さらに, オブジェクトの進化プロセスと, 確率分布モデルのパラメータの変化過程との対応を観測した。本稿では, 短期型開発と長期型開発といった開発方針の違いによるオブジェクトの進化プロセスを定量化した結果得られた確率分布モデルのパラメータの変化過程の差異について議論する。

Empirical Study on Statistically Characteristics of Object Evolution Process

TAKAKO NAKATANI[†] and TETSUO TAMAI^{††}

Distribution of measurements such as the number of lines of code of a class, the number of method definitions of a class, and the number of lines of code of a method, has its median on the left side and a long tail on the right side. Though an overall shape of the distribution pattern is stable and independent of development languages or the way of system evolution, some parameters that characterize the distribution may change over time during the development processes. In this paper, we focus on these changes and examine a causal relationship between the changes and the development processes.

First of all, we try to apply the negative binomial distribution model to the distributions of size data of source code and test goodness of fit. As the results, the distributions fit to the negative binomial distribution model well and change statistical model parameters along the incremental development process. We discuss the relation between development policies, shortsighted or farsighted, and directions of the distribution model parameters changes.

1. はじめに

オブジェクト指向システム開発に適用するメトリクス
の提案が盛んに行われている^{4),18)}。これらの提案では,
実システムに対して計測を行った結果が提示されてい
ることも多く, それらの計測値をもとに作成された頻度分
布の形状には, いくつかの共通点を見ることができる。
たとえば, 計測値の頻度分布が, 左側に最頻値を持ち,
右に尾を引く形状を持つことは, ソフトウェアに適用さ
れる多くの計測値で観測される性質であるし⁶⁾, この形

状には, システムが漸進的に開発されたとしても, その
形状自体が保存されるという性質がある^{12),15)}。

実際の漸進的なシステム開発過程に沿って, クラスや
メソッドといったオブジェクトの定量的な変化過程を詳
細に観測すると, 安定した形状を持つ頻度分布のなかで,
特異な変化を繰り返す少数のオブジェクトを発見するこ
とができる。このようなオブジェクトは, 開発中の要求変
更や設計変更が特定のオブジェクトに集中した際に発見
されやすく, 頻度分布の尾の長さが長くなる現象として
観測される¹⁵⁾。しかし, その変化は, 頻度分布の平均値
に対して, 統計的に有意な差異を与えるものではない¹⁶⁾。

オブジェクト指向システムの漸進的开发で行われる設
計作業は, クラス内部の変更, クラスの継承構造の変更,
クラスそのものの追加/削除に分類される¹⁾。ここで, ク
ラス内部の変更には, インスタンス変数, メソッド, お
よびクラス間の関連の変更が含まれる³⁾。クラスの進化

[†] S Lagoon

SLagoon

e-mail: tina@slagoon.to

^{††} 東京大学大学院総合文化研究科広域科学専攻

Graduate School of Arts and Sciences,

University of Tokyo

e-mail: tamai@graco.c.u-tokyo.ac.jp

という観点から観測すると、クラスに新しい振る舞いを追加するような開発では、クラス内部の変更だけが発生するが、クラスの継承構造を変更する開発では、クラス内部の変更も、実はクラスの抽象化やクラスの分割、統合といった設計作業の副作用であることが多い¹⁵⁾。したがって、オブジェクトの進化プロセスを観測する際は、個々のオブジェクトの変更内容だけでなく、その原因についても観察を行う必要がある。オブジェクトの進化とは、そもそも、オブジェクトを取り巻く利用者要求や開発者の設計意図といった環境の変化への、オブジェクトの適応結果と考えられる¹⁴⁾。本稿では、オブジェクトの進化要因を利用者要求変更へ対応することを主目的とした短期型開発と、再利用性や保守性向上といった将来の開発を意図した長期型開発に分類して議論を進めることにする。

我々は、オブジェクトの進化プロセスを定量的に捉えるために、クラス継承木レベルとシステムレベルの観測を行った。まず最初に、システムレベルで得られる計測値の頻度分布をクラス継承木ごとの層に分割して観測し、得られた分布に確率分布モデルの適用を試みる。左に最頻値を持ち、右に尾を引く離散型の確率分布モデルには、ポアソン分布、幾何分布などがある。理論的な確率分布モデルが1メソッドあたりの行数の頻度分布のような観測結果にどの位適合しているかは、ピアソンの適合度基準を用いて検定する^{9)~11)}。

本稿では、漸進的開発や落水型開発を採用した3システム開発事例で得られた計測結果から、1メソッドあたりの行数が、幾何分布を一般化した負の二項分布に適合することを統計的に検定し、負の二項分布の2つのパラメータの変化過程と、開発型との関係を調査した結果を示す。

本稿の構成は次のようになっている。第2節で調査対象としたシステム開発事例の概要を紹介した後、共通に観測される計測値の頻度分布の傾向を示し、負の二項分布の確率分布モデルへの適合度検定の結果を示す。第3節で負の二項分布の2つのパラメータの変化過程と、パラメータの変化の意味について解析し、両者の対応付けを試みる。第4節で、本研究の考察を述べ、第5節で関連研究を紹介する。

2. 計測調査概要

2.1 計測対象システム概要

計測調査の対象としたシステムについて、その概要を以下に示す。

[シミュレーションエディタ]

- 版数：4
- 計測期間：7か月
- 開発者の人数：1
- 開発言語：Visual Smalltalk
- 開発プロセス：漸進型
- システム規模：Ver.4のクラス数、メソッド数、行数はそれぞれ52, 927, 8677で、Ver.1のそれらの3.25倍、4.80倍、5.56倍であった。

[入金消し込みシステム]

- 版数：4
- 計測期間：7か月
- 開発者の人数：1
- 開発言語：Visual Smalltalk
- 開発プロセス：漸進型
- システム規模：Ver.4のクラス数、メソッド数、行数はそれぞれ62, 2644, 20470で、Ver.1のそれらの1.68倍、3.07倍、3.51倍であった。

[証券管理システム]

- 版数：14
- 計測期間：3か月
- 開発者の人数：4
- 開発言語：Visual Smalltalk
- 開発プロセス：落水型
- システム規模：第14版のクラス数、メソッド数、行数はそれぞれ133, 1487, 14934で、Ver.1のそれらの1.51倍、2.13倍、2.27倍であった。

2.2 計測結果

シミュレーションエディタの4つの版について求めた、1メソッドあたりの行数(MLOC)の基本統計量を表1に示す。入金消し込みシステムおよび証券管理システムにおけるMLOCの基本統計量は表2および表3に示した。各システムを版毎に求めた基本統計量で比較すると、中央値には大きな変化が起きていないことや、その頻度分布が左に最頻値を持ち、右に尾を引く形状となっていることがわかる。

3システムの開発言語であるSmalltalkのプログラマは、メソッドを定義する際にSystemBrowserと呼ばれるツールを使う。このツールには、メッセージ名、コメント行、空行、メソッド本体の4行からなるテンプレートが提供されている。SystemBrowserを使うためって定義されたメソッドの平均的な行数は、ウィンドウ内で一

表1 シミュレーションエディタのメソッド行数の基本統計量

	第1版	第2版	第3版	第4版
平均値	8.08	8.52	9.09	9.36
中央値	5	5	5	5
標準偏差	10.79	16.01	19.53	21.52
最小値	3	3	3	3
最大値	116	261	410	427
メソッド数	193	436	885	927

表2 入金消し込みシステムのメソッド行数の基本統計量

	第1版	第2版	第3版	第4版
平均値	6.78	7.50	7.61	7.74
中央値	4	4	4	5
標準偏差	8.55	8.22	7.93	7.76
最小値	1	1	1	1
最大値	99	99	99	99
メソッド数	861	2491	2705	2644

表3 証券管理システムのメソッド行数の基本統計量

	第00版	第04版	第08版	第12版
平均値	9.41	9.86	10.05	10.11
中央値	4	4	5	5
標準偏差	12.90	13.56	13.67	13.69
最小値	2	2	2	2
最大値	168	178	178	178
メソッド数	698	1406	1466	1466

瞥できる10行程程度の長さになると予想されるが、表1に示すように、メソッドの中には百行を超えるものも存在する。

シミュレーションエディタの計測結果でMLOCの最小値が3となっているメソッドは、本体が1行だけのメソッドである。また、このシステムで最大規模を持っていたメソッドは、システムの版を重ねるたびにその規模を増大させたが、入金消し込みシステムや証券管理システムでは、このような傾向を観測することはできなかった。しかし、いずれのシステムも、その計測値の頻度分布の形状には共通点がある。シミュレーションエディタの各版で得られたMLOCの頻度分布を、それぞれ図1に示し、他の2システムで同様に得られた頻度分布を図2および図3に示す。これらの分布の形状はLorenzらやChidamberらが計測した結果^{2),8)}ともよく似ていることから、言語や開発システムに依存しない性質であると考えることができる。

我々は、これらの分布に対して、ポアソン分布、幾何分布、幾何分布を一般化した負の二項分布などの確率分布モデルを適用することを試みた。事前の調査によって、ポアソン分布は適用できないことが明らかとなったので、ここでは、幾何分布と負の二項分布を適用する。

幾何分布の確率密度関数 $g(x)$ は、成功確率を p 、失敗

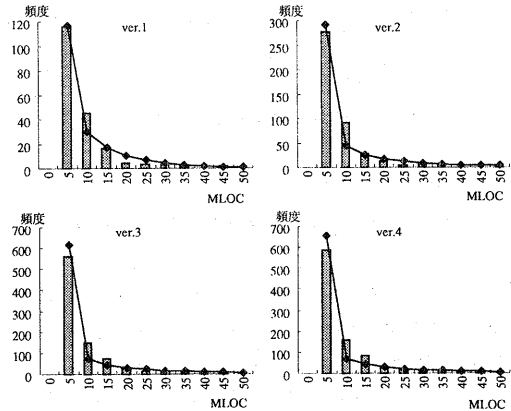


図1 シミュレーションエディタの1メソッドあたりの行数の頻度分布と理論値の比較

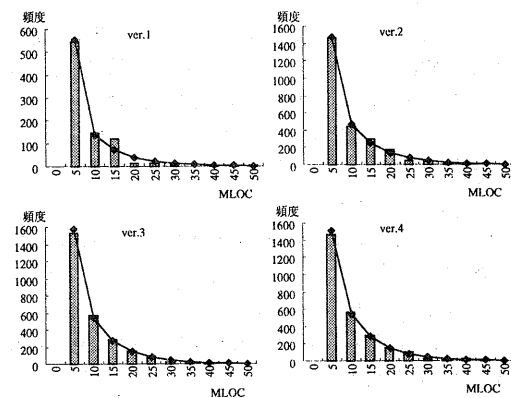


図2 入金消し込みシステムの1メソッドあたりの行数の頻度分布と理論値の比較

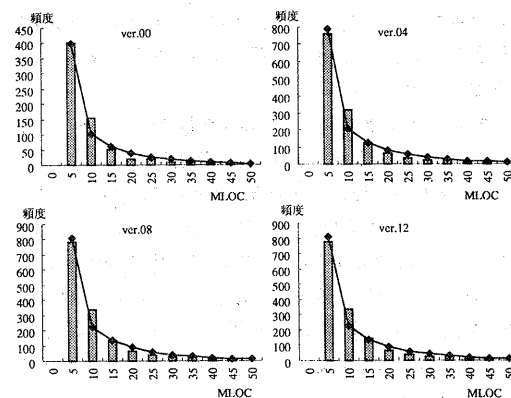


図3 証券管理システムの1メソッドあたりの行数の頻度分布と理論値の比較

確率を $q = 1 - p$ としたとき、最初の成功 S が出現するまでのベルヌーイ試行回数 x を確率変数とすると、 $x = 1, 2, 3, \dots$ で、

$$q(x) = pq^{x-1}, \quad x = 1, 2, 3, \dots$$

と与えられる。幾何分布の期待値と分散は、それぞれ次のようになる。

$$E(X) = 1/p, \quad V(X) = q/p^2$$

幾何分布を MLOC の頻度分布にあてはめて解釈すると次のようになる。

「プログラムの各行が終了行となる確率 p は一定であるとき、開発者が無作為にプログラム行の集合から 1 行ずつ逐次選択し、その行が終了行であれば成功、それ以外は失敗と考える。行の選択は、選択された行が終了行となるまでの選択回数を確率変数とする分布である。ここでは、プログラム行の選択回数が 1 メソッドあたりの行数となる。」

負の二項分布の確率密度関数 $f(x)$ は、成功確率を p 、失敗確率を $q = 1 - p$ としたとき、 r 回目の成功 S が出現するまでのベルヌーイ試行回数 x を確率変数とする。したがって、 $x = r, r + 1, r + 2, \dots$ のもとで、確率密度関数 $f(x)$ は、

$$f(x) = {}_{x-1}C_{r-1} \cdot p^r \cdot q^{x-r},$$

と与えられる。負の二項分布の期待値と分散は、それぞれ次のようになる。

$$E(X) = rq/p, \quad V(X) = rq/p^2$$

負の二項分布を MLOC の頻度分布にあてはめて解釈すると次のようになる。

「1 メソッドが完了するまでには、何回かの成功事象が発生しなければならないと考える。各行は一定の成功確率 p を持つため、開発者が無作為にプログラム行の集合から 1 行ずつ逐次選択し、規定の回数だけ成功事象が発生するまで繰り返される試行回数を確率変数とする分布である。」

2.3 確率分布モデルとの適合度検定

3 システムで求めた MLOC の頻度分布は、幾何分布よりも負の二項分布から求めた理論値によく適合していた。図 1、図 2 および図 3 に、折れ線グラフで負の二項分布から求めた理論値を示す。ここで、計測値の負の二項分布への適合度を検定する。確率分布モデルとの適合度検定には、5% の有意水準（片側）でピアソンの適合度基準を用いる。ここで提起する帰無仮説 H_0 と対立仮説 H_1 は次のようになる。

$$H_0: p_1 = p_{10}, p_2 = p_{20}, p_3 = p_{30}, \dots, p_k = p_{k0}$$

$$H_1: \text{少なくとも 1 つの } p_i \neq p_{i0}$$

ただし、階級 i の観測確率を p_i とし、理論確率を p_{i0} とした。検定を行うにあたり、行数の小さいメソッドでは、空行の有無、コメント行の有無などのプログラムの嗜好が分布に大きな影響を与えるため、データ区間を 5 行に設定して求めた頻度分布を検定対象とする。

2.3.1 検定手順

各システムの各版に対して適合度検定を試みたところ、帰無仮説を棄却しなければならなかった。この原因は、計測した頻度分布が負の二項分布には従わないか、もしくは、頻度分布が異なる性質を持つ複合分布となっていることが考えられる。図 1、図 2 および図 3 では 5 行ごとに区間を区切っているのでわかりにくいですが、1 行毎の頻度分布では、ヒストグラムにいくつかの凸凹を観測することができる。これは、このヒストグラムの対象となっている標本が、異なる性質を持つ標本の集まりであることを示唆している。

そこで、適合度検定を次の手順で進めることにした。

- (1) 標本をヒストグラム上で凸凹が生じない、いくつかの層へ分類する。
- (2) 各々の標本に対して平均値と分散を求める。
- (3) 求められた平均値と分散から負の二項分布のパラメータを推定し、理論値を求める。
- (4) 理論値と計測値との差分を求め、適合度検定を行う。

標本を同じ性質をもつと思われる層に分類するにあたり、クラス継承木による分類を適用する。オブジェクト指向システムのクラス継承木は、個々に固有の 1 メソッドあたりの行数の値を保持しているため¹⁶⁾、分布の平均値によってパラメータが左右される負の二項分布に適合する層を分類する目的に適していると判断したためである。

2.3.2 検定結果

適合度検定の結果、いくつかの適合しない標本の集合は存在するが、帰無仮説を棄却できないことが統計的に確認できた。

いくつかの標本が理論値に適合しなかった理由として考えられる事項を次に挙げる。

- データ区間の境界上で生じた誤差の影響とみなせる不適合。これは、データ区間の境界 ± 1 行の頻度が検定結果に大きな影響を与えている標本で発生した。
- クラス継承木による層別だけでは頻度分布の凹凸を解消できなかったために発生した不適合。これは、クラス継承木に着目したクラスの分類のなかに、手続き指向的なクラスとオブジェクト指向的なクラスといった、性質の異なるクラスが混在する設計で観測された。この複合分布状態を解消するために、より

粒度の小さい層へ分割することを検討したが、標本数が少なくなってしまう頻度分布の議論を行う意味がなくなった。

3. 計測結果の解析

3.1 負の二項分布の性質

負の二項分布の確率密度関数 $f(x)$ に従う標本の期待値と分散は、 $\mu = r/p$ および $\sigma^2 = rq/p^2$ で与えられる。

そこで、計測値から求められた平均値 \bar{X} と分散 s^2 から、負の二項分布のパラメータ p と r を次の式から推定する。

$$p = \bar{X} / (s^2 + \bar{X}), \quad r = \bar{X}^2 / (s^2 + \bar{X})$$

$\bar{X} = r/p$ であるから、横軸に p 、縦軸に r をとった平面上に計測値から推定した負の二項分布のパラメータをプロットすると、クラスレベルの進化をパラメータに対応する点の動きとして観察できるようになる。

この平面上の点 (p_1, r_1) から原点までの距離 l は、

$$l^2 = p_1^2 + r_1^2 = (\bar{X}^2 + \bar{X}^4) / (s^2 + \bar{X})^2$$

で与えられる。標本はクラス継承木ごとに層別されているので、分布の平均値 $\bar{X} = r/p$ は一定と考えてよいから、クラス継承木の進化から得られる点系列は、平均値の傾きをもった直線上を移動するはずである。

さらに、 \bar{X} 一定のもとでは、クラス継承木の進化プロセスで得られた標本の各点と原点との距離は、標本の分散の変化と解釈することができる。すなわち、点が原点に近づくということは標本の分散が大きくなることを意味し、点が原点から遠ざかる方向に移動するということは、標本の分散が小さくなることを意味する。

分散が大きくなるのは、規模の大きいメソッドが追加されたり、既存のメソッドの規模が拡大したときである。このような現象は、開発者が短期型開発を行った場合に発生するだろう。また、分散が小さくなるのは、クラスの抽象化やメソッドの分解を行って、クラスの規模やメソッドの規模を平均化したときである。このような現象は、開発者が長期型開発を行った場合に発生するはずである。

3.2 進化プロセスの観測

図4、図5、図6には、3システムで求められた1メソッドあたりの行数(MLOC)の分布から推定した負の二項分布のパラメータの値をプロットした。

シミュレーションエディタは、ver.1からver.2にかけて変化が大きく、点の系列は右上から左下に移動しているが、入金消し込みシステムでは、左下から右上に点系列が移動している。証券管理システムでは、点系列の動きが少なく、問題領域に関するクラス継承木で、開発初

期の1か月間(ver.00-ver.04)で右から左に向かう大きな点の動きを観測した。各システムの開発時期と開発型を開発者へのインタビューや定性的な設計情報を解析しながら、点の動きを関連づけて観測をしたところ、負の二項分布のパラメータの動きから想定された各クラス継承木の開発過程と矛盾していないことが明らかとなった。

各システムの観測結果を以下にまとめた。

● シミュレーションエディタ

ver.3まで、このエディタは、単純な機器構成のエディタから、機器の増加、本格的なシミュレーション機能を持ったエディタへと成長した。このシステムの開発過程の特徴は、初期に開発した単純なプロトタイプを利用者要求に適應させるために3回の漸進的な開発をし続けた点にある。開発者は、各版を毎月顧客に提示し、顧客が試用した後、要求変更を受け取り、次の納期までに要求変更に対応するといったスケジュールで開発を進めた。この間、クラスの分割、クラス階層の変更も発生しているが、開発者はインタビューの中で、新たな要求への対応を開発作業の中心として挙げ、再利用部品の開発や可読性向上、保守性向上のための開発作業については、納期の問題から十分検討できなかったと述べている。以上の調査から、点系列が原点に向かって移動している現象は、短期型開発が主に行われたことを表していると解釈できる。

● 入金消し込みシステム

ver.1を顧客に提示した後、要求変更を受け取りながら開発が進められた点ではシミュレーションエディタと同様である。異なる点は、納期の間隔と、ver.1の時点で、システム全体のフレームワークが決定されており、ver.4に至るまで、このフレームワークを変更しなければならないような要求変更が発生しなかった点にある。開発者によると、ver.2からver.3への拡張は、画面の項目表示位置の変更や、消し込み操作後のデータベース更新のタイミング、メニューの許可/不許可の調整といった使い勝手に関する要求がほとんどを占めていた。さらに、この開発過程で、クラスへの機能追加は若干あったが、それ自体がクラス階層の設計変更を起すものではなかったと述べていた。

ところが、クラスレベルの進化プロセスを詳細に観察すると、13クラスの名称が変更されていることがわかる。これらのクラスは、入金消し込みシステム向けのアプリケーション依存のクラスとして開発さ

れ、その後、再利用可能クラスとして設計変更されて開発されたクラスである。再利用可能クラスとなるにあたり、クラスの抽象化、継承構造の変更、クラスの分割/統合が行われていた¹⁴⁾。

以上の調査から、点系列が原点から離れる方向へ移動している現象は、短期型開発を進めながら、長期型開発も行われ、全体の傾向としては長期型開発を重視して進められていたことを表していると解釈できる。

● 証券管理システム

このシステムでは開発中に利用者の要求変更は発生していない。また、開発中のクラス継承構造の組み替えや、新たなクラスの追加といった仕様変更も発生していない。

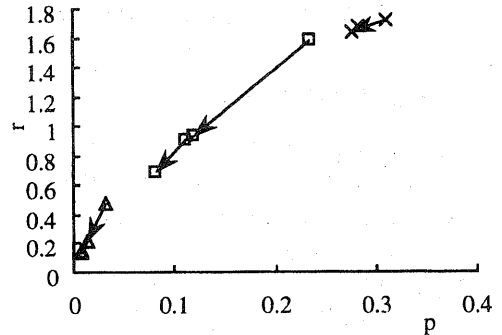
点の動きを観測すると、開発初期の1ヶ月で原点方向に移動し、その後変化が起きていないが、これは、1ヶ月後からほぼコーディング作業を終了し、その後も、開発されたクラスの仕様を整理するといった長期型開発は行われていなかったことを表わしていると解釈できる。この解釈は、開発者のインタビューで確認した内容とも一致した。

● 全システムを通して

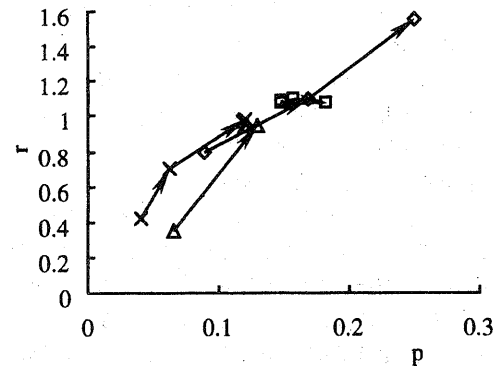
図4、図5、図6から、各システムの点系列がほぼ直線上に乗っていることがわかる。これは、層別する際に選択したクラス継承木の性質「MLOCの平均値はクラス継承木ごとに一定である¹⁶⁾」を表している。クラス継承木において、その進化プロセスでは、分散に統計上有意な差異は認められなかったが¹⁵⁾、確率分布モデルを適用することによって、分散の差異を、モデルのパラメータの差として観測できたことになる。

4. 考 察

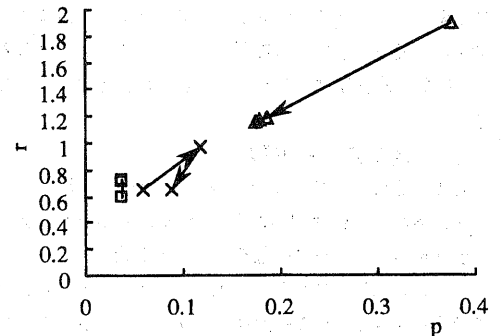
本稿では、1メソッドあたりの行数の計測結果から求めた頻度分布をクラス継承木に層別し、負の二項分布の適合度を検定した。また、負の二項分布のパラメータの変化が表す意味について議論し、実際の計測結果とシステムの定性的な解析、開発者へのインタビューを通し、パラメータの変化が表す開発過程の特徴について議論してきた。さらに、横軸に負の二項分布の成功確率 p をとり、縦軸に成功回数 r をとった平面上に、それぞれの頻度分布から推定した p および r をプロットすると、各点が、次に示す性質を持ちながら開発過程に従って変化することを視覚的に観測できるを明らかにした。



□ 表示クラス × 計算クラス △ 編集クラス
 図4 シミュレーションエディタのクラス継承木で層別したMLOCの分布パラメータの変化



◇ 一覧表操作クラス □ 表抽出クラス
 △ 永続化クラス × 仲介クラス
 図5 入金消し込みシステムのクラス継承木で層別したMLOCの分布パラメータの変化



□ 仲介クラス × 定数表クラス △ 証券領域クラス
 図6 証券管理システムのクラス継承木で層別したMLOCの分布パラメータの変化

- (1) クラス継承木ごとに点系列は直線上の乗る。
- (2) 短期型開発の場合、点系列は原点に向かって移動する。
- (3) 長期型開発の場合、点系列は左から右へ移動する。

上記の結論は、 p , r の性質から、

- 短期型開発では、成功事象の確率 p が下がり、メソッド完結までに要する行数が減る
- 長期型開発では、成功事象の確率 p が上がるが、メソッド完結までに要する行数も増える

ことを表わしている。成功事象の確率 p と成功回数 r を、それぞれ、メソッドを完結させるために要求される作業制約の強さと成功事象と考えられる作業量を表わすと仮定すると、それぞれの開発型に対応して、次のような解釈を行うことができる。

たとえば、厳しい作業制約がなければ、成功確率はそれほど高くないが、作業制約に沿った検討を行う必要もないため、必要な作業量、すなわちプログラミング終了までに要求される成功回数は減ることになる。逆に、成功事象の確率を上げるための作業制約が詳細に提供されていれば、プログラマは的確なプログラミングを行えるようになるが、そのために必要な作業量、すなわち成功回数は増えることになる。すなわち、短期型開発では、要求を満足させることが第一の目標となるため、プログラミングの作業方針は厳しくないため、作業の制約は緩いと考えられるが、長期型開発では、継承構造の組み替えや、クラスの複雑度などの調整が進むにつれて、プログラミングの作業方針や設計指針が提供され、それがプログラミングの制約となり、プログラミング時に考えるべき事柄が増大すると考えられる。この解釈は、短期型開発や長期型開発で行われるプログラミング作業の直観的な解釈とよく合致するのではないだろうか。

このように、2種類の開発型で観測された負の二項分布の p および r の軌跡は、平均値や分散といった確率分布モデルの性質の変化を表わしているだけでなく、それぞれの開発でプログラマに課せられる制約の強さや作業量とみなすことも可能である。

5. 関連研究

Lorenz らは、ある計測値が、一般的なオブジェクト指向システムで観測される平均値から大きく外れた値である場合、その計測値は不適切な設計を表わしている可能性が高いので、メンターと相談して設計を検討し直すようにと述べている⁶⁾。しかし、システムには、それぞれ固有の開発形態や開発指針があり、一般的な値と比較す

ることで設計を評価するのは適切ではない。

著者らはこれまでの研究の中で、システムごとの開発過程の成果物を計測対象とし、成果物の定量的な差に注目して設計の評価を行うことを検討してきた。その結果、メソッドの規模やクラスの規模が巨大であるという理由だけで、設計上の問題点を示唆することはできないことを指摘した^{12), 15)}。たとえば、複雑な物理計算を行うメソッドの行数が、1メソッドあたりの平均行数に比べて大きくなったとしても、このようなメソッドを分割する必要はない。問題なのは、大規模メソッドが要求変更の影響を受けて増大化し続けることである。オブジェクトの進化プロセスの調査から設計上の問題点を示唆できるのは、計測値の分布がシステムの成長によらずに変化しないという傾向を根拠にしていた。平均値や中央値が大きく変化しないオブジェクトは、1メソッドあたりの行数の他に、1クラスあたりのメソッド数、行数などがある。しかし、本研究によって、計測値の分布に確率分布関数を適用することで、分布の変化過程を分布モデルのパラメータの変化過程として観測できることが明らかとなった。

プログラム進化では、Belady と Lehman の研究が有名である⁵⁾。彼等は、システムレベルの観測によってプログラムの進合法則を導いた⁷⁾。オブジェクトの進化も、システム全体の開発過程と対応付けて議論するためには、システムレベル、あるいはクラス継承木レベルで観測を行う必要がある。たとえば、新たな仕様変更に対応するために、それなりの設計方針があるだろうし、長期に渡る継続的な開発が要求されている場合は、システムの将来の進化にあわせた拡張性を考慮した設計が望まれる。我々は、このような開発方針の影響を大きく受けると考えられるメソッドやクラスの進化をクラス継承木レベルで観測した。プロジェクト管理者にとって、オブジェクトの進化は、個々のオブジェクトがどのような進化プロセスを経て現在に至っているかといったマイクロ情報だけでなく、担当者がプロジェクトに適切な開発方針を守りながら開発を進めているか否かを示すマクロ情報も重要であろう。本研究では、計測された分布のパラメータが、長期型と短期型の2種類の開発形態によって、異なる変化過程を持つことを示すことができた。

玉井と鳥光は、世代に渡るシステムの進化プロセスについて調査を行い、必ずしも寿命の長いシステムが開発の目標ではないという実態を明らかにした¹⁷⁾。本研究では、3つのシステムの計測結果をもとに、システムの成長とクラス継承木全体の進化プロセスとを調査するにと

どまった。しかし、図4のように、このまま原点に近付いて行くと、システムの寿命はどのようになるのかが知りたくなる。シミュレーションエディタは、その後、仕様の見直しが行われ、第二世代のシステムが開発された。したがって、このデータを、ひとつのシステムの終焉とみなすことができるかも知れない。今後は、本研究で得られた結果の検証を進め、一般性を高める研究を進めていきたい。

6. ま と め

本稿では、実開発システム事例から得られた1メソッドあたりの行数の頻度分布に負の二項分布を適用し、そのパラメータの変化過程と開発の型を関連づけて解析した。開発によって1メソッドあたりの行数といった値の分布が、クラスの設計変更、クラス継承木の設計変更、システム全体の開発の進捗といった現象を反映しており、分布のパラメータの変化を観測することによって、開発の制約の程度を知ることができることを示した。

参 考 文 献

- 1) Banerjee, J., Kim, W., Kim, H. J. and Korth, H. F. : "Semantics and Implementation of Schema Evolution in Object-Oriented Databases," *Proc. of the ACM SIGMOD Annual Conf. on Management of Data*, pp. 311-322 (May, 1987).
- 2) Chidamber, S. R. and Kemerer, C. F. : "A Metrics Suite for Object Oriented Design," *IEEE Transactions on Software Engineering*, Vol.20, No.6, pp. 476-493 (1994).
- 3) Erradi, M., Bochmann, G. V., and Dssouli, R. : "A Framework for Dynamic Evolution of Object-Oriented Specifications," *Proc. of the 11th Conference on Software Maintenance*, IEEE, pp. 96-104 (1992).
- 4) Henderson-Sellers, B. : *Object-Oriented Metrics - Measures of Complexity*, Prentice Hall (1996).
- 5) M. M. Lehman. and L. A. Belady. : *Program Evolution*, Academic Press (1985).
- 6) Kitchenham, B. A. and Linkman, S. J. : "Design Metrics in Practice," *Information and Software Technology*, Vol. 32, No. 4, pp. 304-310 (1990).
- 7) Lehman, M. M. : "Programs, Life Cycles and Laws of Software Evolution," Lehman, M. M. and Belady, L. A. edit. *Program Evolution*, Academic Press, pp. 393-449 (1985).
- 8) Lorenz, M. and Kidd, J. : *Object-Oriented Software Metrics*, Prentice Hall (1994).
- 9) 養谷千風彦, : 統計学のはなし, 東京図書 (1987).
- 10) 養谷千風彦, : 統計学入門 1,2, 東京図書 (1994).
- 11) 東京大学教養学部統計学教室編, : 統計学入門, 東京大学出版会 (1991).
- 12) 中谷多哉子, 友枝敦, 酒匂寛, 玉井哲雄, : "オブジェクト指向システムの進化プロセスの定量的分析," 日本ソフトウェア科学会第13回大会論文集, pp. 389-392 (1996).
- 13) 中谷多哉子, 玉井哲雄, 友枝敦, 酒匂寛, : "オブジェクト指向によるシステムの進化を表わすメトリクスの検討," ソフトウェアシンポジウム'96論文集, SEA, pp. 52-62 (1996).
- 14) Nakatani, T., Tamai, T., Tomoeda, A. and Matsuda, H., : "Towards Constructing an Object Evolution Model," *Proc. of the Asia-Pacific Software Engineering Conference '97*, IEEE, Hong Kong, pp. 131-138 (1997).
- 15) Nakatani, T., Tamai, T., Tomoeda, A. and Sakoh, H., : "Quantitative Analysis on Evolution Process of Object-Oriented Systems," *Proc. of the International Symposium on Future Software Technology '96*, Xi'an, China, pp. 49-56 (1996).
- 16) Nakatani, T., Tamai, T., : "Evolutional Characteristics of Class Inheritance Trees," *Proc. of the International Symposium on Future Software Technology '97*, Xiamen, China, pp. 44-51 (1997).
- 17) Tamai, T. and Torimitsu, Y., : "Software Lifetime and its Evolution Process over Generations," *Proc. of the Conference on Software Maintenance*, pp. 63-69 (1992).
- 18) Whitty, R. : "Object-Oriented Metrics: A Status Report," *Object Expert*, Vol.1, No.1, pp. 35-40 (1996).