

第 2 章

システムレベルから観測した進化過程

2.1 概要

本章では，オブジェクトの進化過程を利用者環境変化および技術的環境変化と関連づけた分析結果について議論する．

システムレベルの分析では，クラスの行数やメソッド数，メソッドの行数といった計測値の分布が，左に最頻値を持ち右に尾を引く形状を持っており，システム進化によらず，その形状は大きく変化しないことが明らかとなった．また，この分布には，例外的な規模を持ち，他のオブジェクトに比べて早く進化するオブジェクトも存在していた．このようなオブジェクトは，開発者が問題箇所として注目し，設計変更の対象として抽出していたオブジェクトと一致した．以上の調査から，計測値の分布から特異な進化を示すオブジェクトを発見すれば，設計変更候補のオブジェクトを抽出できることが明らかとなった．

さらに，複数のクラスが連動して進化する過程を，開発者の設計意図と関連づけた．本章では，これらのオブジェクトの計測値について，頻度分布の安定性を議論したあと，進化過程と設計変更事由および設計変更による効果を対応づけながら議論する．

2.2 計測値の頻度分布の安定性について

図 2.1 に SystemA に定義されたメソッドの行数 (MLOC) について，その頻度分布と累積度数分布をシステムの版ごとに示し，表 2.1 にその基本統計量を

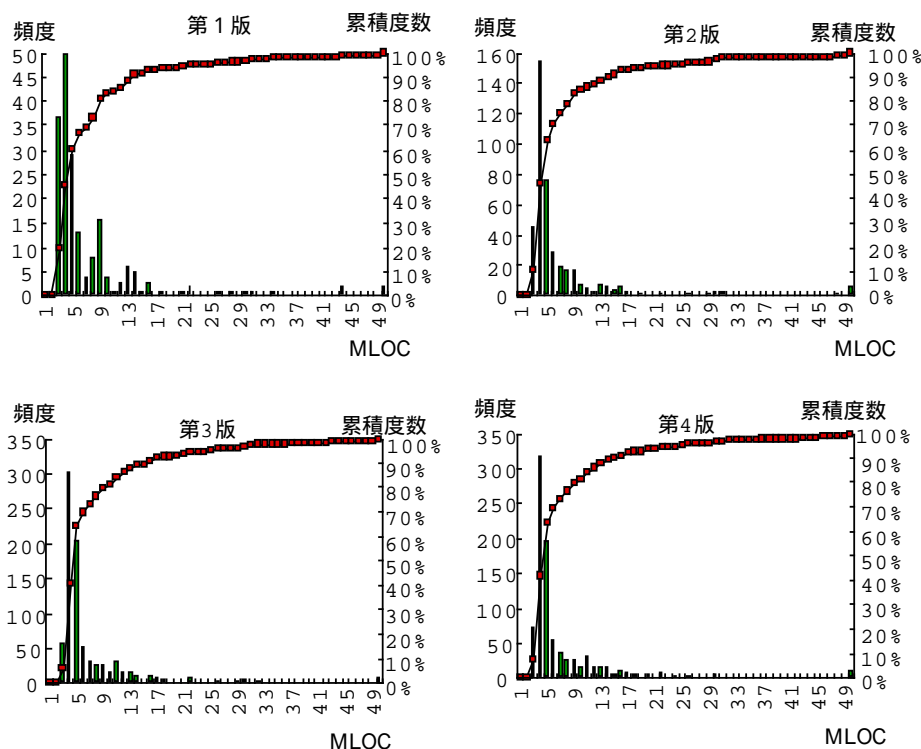


図 2.1: SystemA のメソッド行数 (MLOC) の頻度分布と累積度数分布

示す。SystemA の各版の分布を図 2.1の左上, 右上, 左下, 右下にそれぞれ示した。

各版を比較すると, 中央値には変化が起きていないことがわかる。本研究で調査を行った 3 システムの開発言語である Smalltalk は, メソッドを定義するためのテンプレートを提供している。テンプレートは, 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。SystemA 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, Smalltalk では, プログラマがメソッドを定義する際に SystemBrowser と呼ばれるエディタを使う。このエディタを使うと, 平均的なメソッド行数はウィンドウ内で一瞥できる 10 行程度の長さになると予想される。しかし, 表 2.1 に示すように, メソッドの行数の最大値が, 数百行となるものもあった。また, 進化過程の調査から, このメソッドは版を重ねるたびに行数を増大させていることもわかった。

表 2.1: SystemA 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 2.2: SystemB 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

SystemB および SystemC におけるメソッド行数の基本統計量を表 2.2 および表 2.3, 表 2.4 に, その頻度分布を図 2.2 および図 2.3 に示す. いずれの例も, SystemA と同様, 中央値には大きな変化はなく, 分布の形状も安定している.

これらの分布の形状は Lorenz らが計測した結果ともよく似ている [37].

ここで, これらの分布に統計モデルをあてはめることを試み, 分布の形状が表す意味について考えることにする. 適用を試みた統計モデルは幾何分布である. 幾何分布の確率 $f(x)$ は, 成功確率 p とし, 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,

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

で与えられる. ただし, 失敗確率 $q = 1 - p$ とする. 幾何分布をメソッド行数

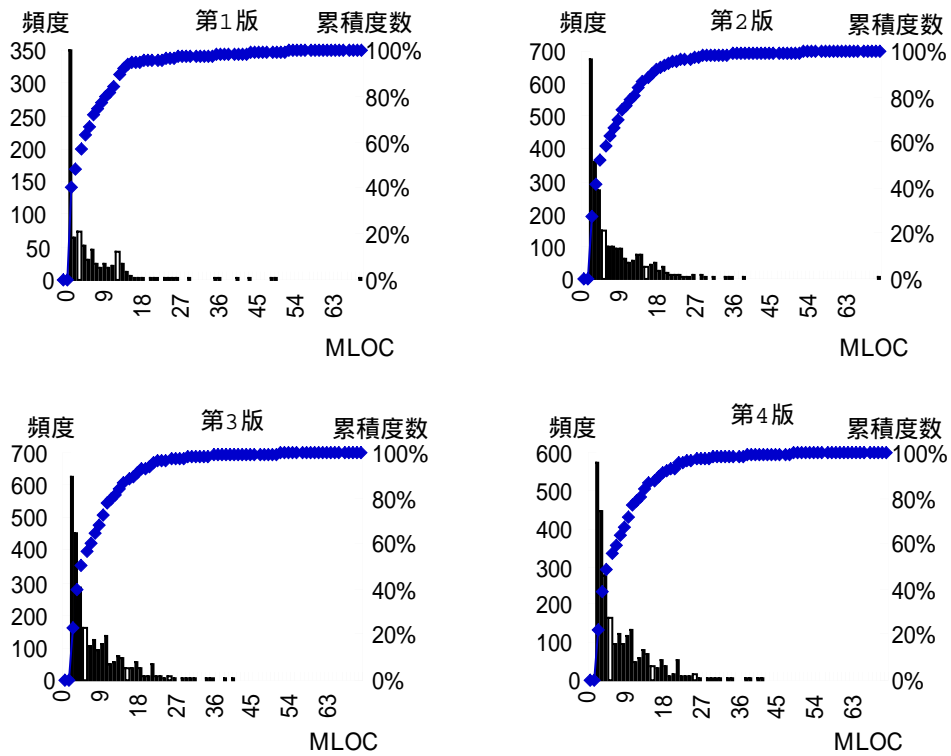


図 2.2: SystemB のメソッド行数 (MLOC) の頻度分布と累積度数分布

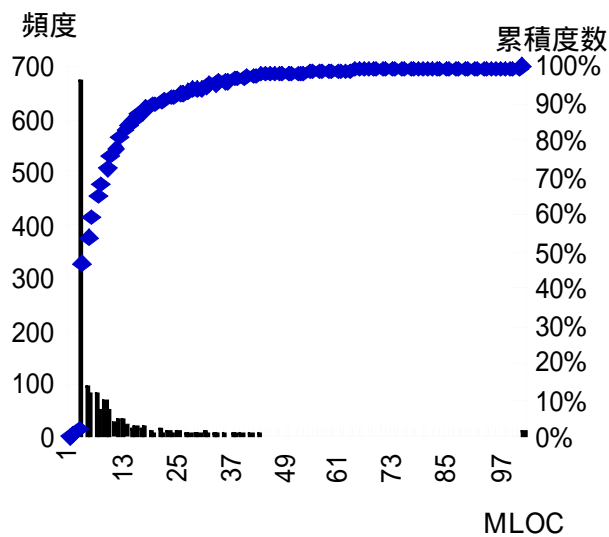


図 2.3: SystemC のメソッド行数 (MLOC) の頻度分布と累積度数分布

表 2.5: 3 システム全体で定義されたメソッド行数の基本統計量

平均値	標準偏差	最小値	最大値	メソッド数
8.15	11.79	1	427	12627

の頻度分布にあてはめると「メソッドの行を先頭から順に検査したとき，最初にメソッドの終端が出現するまでの行数 x の確率がメソッド終了確率 $p = 1/(\text{メソッドの平均行数})$ の幾何分布に従う」と解釈できる．図 2.4 に本研究の対象 3 システムで定義された全メソッドの行数を標本とする頻度分布と累積度数分布を表わした．表 2.5 にこの標本の基本統計量を示す．頻度分布の階級は，メソッド行数の階級を 2 行ごとに区切って表した．図の折れ線グラフは，標本の平均値をパラメータとする幾何分布から頻度分布を求めて，理論値として表わしたものである．

直観的には，メソッド行数の分布が理論的に求めた幾何分布によく従っているように見える．

メソッド行数の分布が幾何分布に従っていることを統計的に検定するために，5% の有意水準を用いてピアソンの適合度検定を行った．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．

$$H_0: p_1 = p_{1_0}, p_2 = p_{2_0}, p_3 = p_{3_0}, \dots, p_k = p_{k_0}$$

$$H_1: \text{少なくとも 1 つの } p_i \neq p_{i_0}$$

ただし，階級 i の観測確率を p_i とし，理論確率を p_{i_0} とした．ピアソンの適合度検定では， n 回の試行（この場合は全メソッド数=12627）において，以下の式を用いた検定を行う．

$$\text{適合度検定: } \chi^2 = \sum_{i=1}^k \frac{(np_i - np_{i_0})^2}{np_{i_0}} \sim \chi^2(k-2)$$

上記の χ^2 の値が k を階級の数とした $\chi^2(k-2)$ の値よりも小さければ帰無仮説を採択し，大きければ帰無仮説を棄却する．検定の結果，4 行以下の観測頻度と理論値との差が大きく帰無仮説を棄却せざるを得なかった．観測結果から求めた頻度分布が幾何分布に従わない要因として，

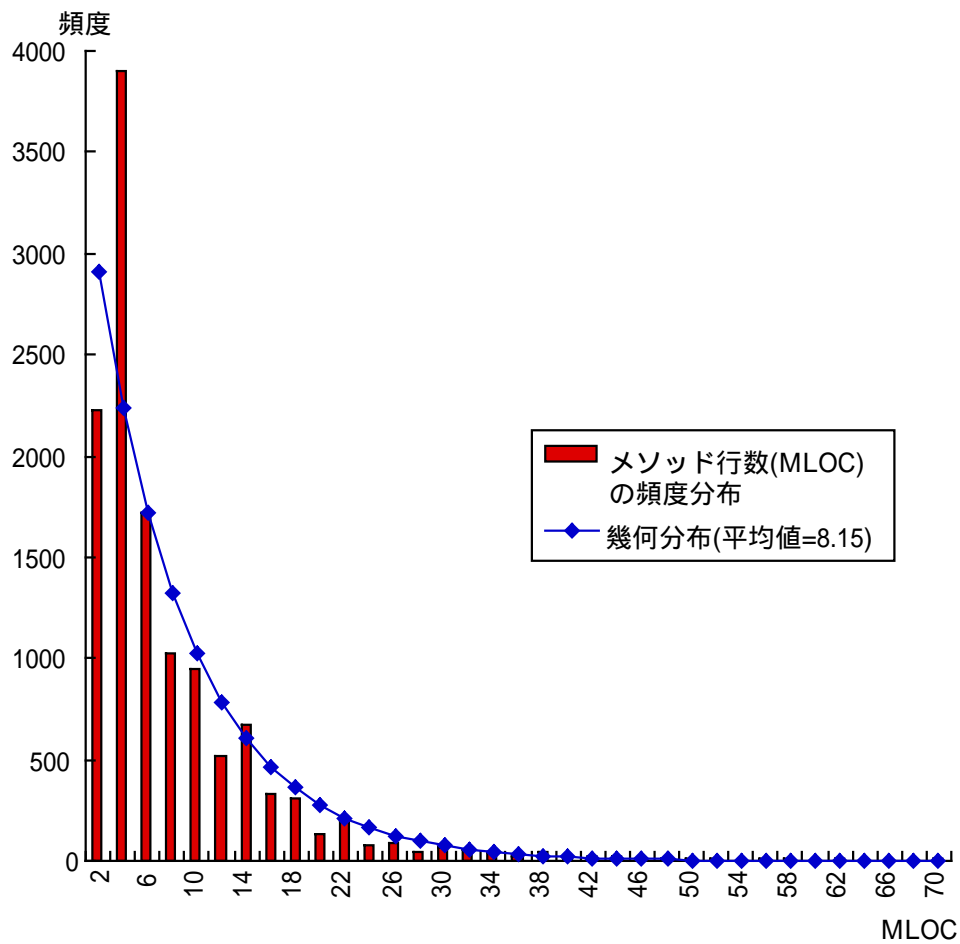


図 2.4: 3 システム全体で定義された全 12627 メソッドを対象としたメソッド行数 (MLOC) の頻度分布と幾何分布による理論分布の比較。幾何分布のパラメータは、観測データから求めた平均値=8.15 を用いた。

- 行数の小さい部分では，1 行の本体のメソッドでも，空行の有無，コメント行の有無などのプログラマの嗜好が分布に大きな影響を与える．
- 検定で用いた幾何分布は，すべてのメソッドが x 行で終了する確率を等しいと仮定する．しかし，ここで集計したメソッドの中には，仕様上の複雑度が異なるメソッドも含まれており，仕様の記述が終了するために要する行数の平均値は，全メソッドで一定となるとは考えられない．これは，すべての行における終了確率が一定であるとする幾何分布の前提条件を満足しない．

といった要因を考えることができる．

検定の結果，帰無仮説は棄却されたが，メソッドの行数の分布が幾何分布に近い分布となることを確認することはできた．この研究成果は，メソッドの平均行数を見積もれば，規模の大きいメソッドの割合を推定できることを示唆している．

2.3 技術的環境変化と進化の関係

2.3.1 SystemA の進化

新たなクラス継承木の出現やクラスの分割は，将来の開発のためにクラスの仕様を整理するために行われる設計変更であるから，システムレベルにおけるオブジェクトの技術的環境への適応と考えることができる．また，クラスの名称変更も，利用者環境の変化によって開発者自身が問題領域への理解を深めるという技術的環境変化への適応である．

図 2.1 では，分布の右端に現われる少数のメソッドを観測した．これらのメソッドは開発当初から定義され，第 4 版までの全期間にわたって存在し続けたメソッドである．行数の変化を追跡調査した結果，版を増すごとに，これらのメソッドはその行数を増加させていることがわかった．開発者は，このようなメソッドの存在について次のように述べている．

「巨大なメソッドが，システムの成長に伴って規模を増大させたのは，偶然ではなく，要求変更をこれらのクラスに集中させてシステム全体に及ぶ設計変更を避けたためである．メソッドの規模が増大したのはフレームワークに原因

があった。フレームワークの設計上の問題には第3版の要求変更を受け取った時点で気付いたが、システム全体のフレームワークを変更する時間的な余裕はなかったため、小数のメソッドで要求変更に対応する設計方針を選択した。」

第3版では、設備の種類が増加と、熱交換のための冷却媒体の流れを制御する機構の変更が要求された。巨大なメソッドを保持していたクラスは、様々な設備との熱交換条件を保持し、設備の配置に従って熱交換を制御する部品である。第3版で、このクラスに冷却媒体の流量などを制御するための条件を追加している。同じ要求追加に対しては、流量を制御するための役割分担を他の設備オブジェクトにも与えて対処する選択肢もあったが、開発期間の制約によって、変更内容を一箇所に集中させる戦略が選択されていた。

以上の観測とインタビュー結果から、定量的に特異な進化を示すメソッドには、設計上の問題がある可能性の高いことが明らかとなった。

定量的に特異な進化をするクラスについて、開発者はどのような設計意思決定を行っているのか。SystemAで行われた設計変更を、クラス継承木の進化から観測した。

SystemAは、次に示す3つの主なクラス継承木から構成されている。

- 編集クラス木：シミュレーションに必要な初期値を与える利用者インタフェースのクラス群。
- 表示クラス木：シミュレーションの設備を表わすアイコンなど、設備を視覚的に組み立てる利用者インタフェースを提供するクラス群。設備の組み立てに関する制約条件も保持する。
- 計算クラス木：シミュレーションの計算エンジンを提供するクラス群。

表2.6にはクラス継承木を構成するクラスの数(NCBT)の変化の軌跡を示す。第2版では、計算クラス木が新たに追加されている。第2版の要求追加では、熱交換制御のためにいくつかの熱交換アルゴリズムが提供された。そのため、開発者は、将来の熱交換アルゴリズムの複雑化に対処するために表示クラス木のクラスから熱交換の計算部分だけを取り出して新たなクラスを構築した。こうして形成されたクラス継承木が計算クラス木である。第2版以降、個々のク

表 2.6: クラス継承木を構成するクラスの数 (NCBT) から観測したクラス継承木の進化過程

クラス木名	第 1 版	第 2 版	第 3 版	第 4 版
編集クラス木	3	5	10	11
表示クラス木	6	8	16	18
計算クラス木	-	6	14	16
物理定数保持クラス木	5	5	5	5
合計	16	26	47	52

ラス継承木には新たな子継承木やクラスも追加されているが、新たなクラス継承木は追加されていない。

それでは、SystemA における技術的環境の変化と、定量的なクラスの進化過程との関係について分析を進めよう。図 2.5 にクラス継承木の構造が変化した様子を示す。

この図には、次の 2 つの群に属するクラスが表示されている。

- 第 1 群：クラスの分割に関わったクラス群（第 1 版から第 2 版の間起こった表示クラス木の分割）
- 第 2 群：クラス継承木の中でスーパークラスの組み替えに関わったクラス群

図 2.6 に SystemA のクラスのメソッド数から観測した変化の軌跡を示す。図で示された折れ線は、それぞれクラスの進化の軌跡を表す。グラフの横軸はシステムの版番号を表し、縦軸はメソッド数を表す。

1. 第 1 群のオブジェクトの進化：クラス分割

SystemA の第 1 版で最大の規模（行数とメソッド数）を持っていたクラスは表示クラス木に属するクラス P5 である。開発者は、第 2 版の要求追加によって、表示クラス木を構成していたすべてのクラスを分割し、熱交換の計算を実行する計算クラス木を新たに定義した。このような設計

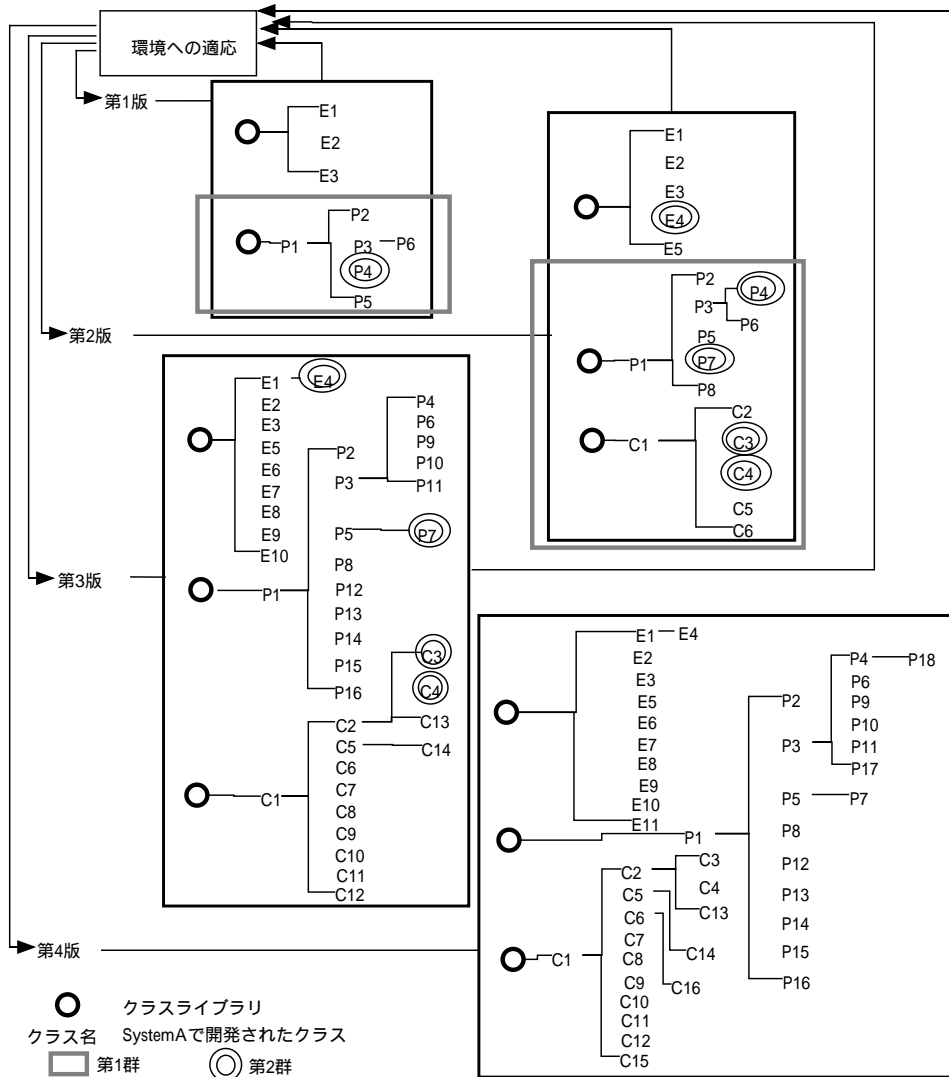


図 2.5: SystemA のクラス継承木の進化過程 (E_i :編集クラス木, P_i :表示クラス木, C_i :計算クラス木)

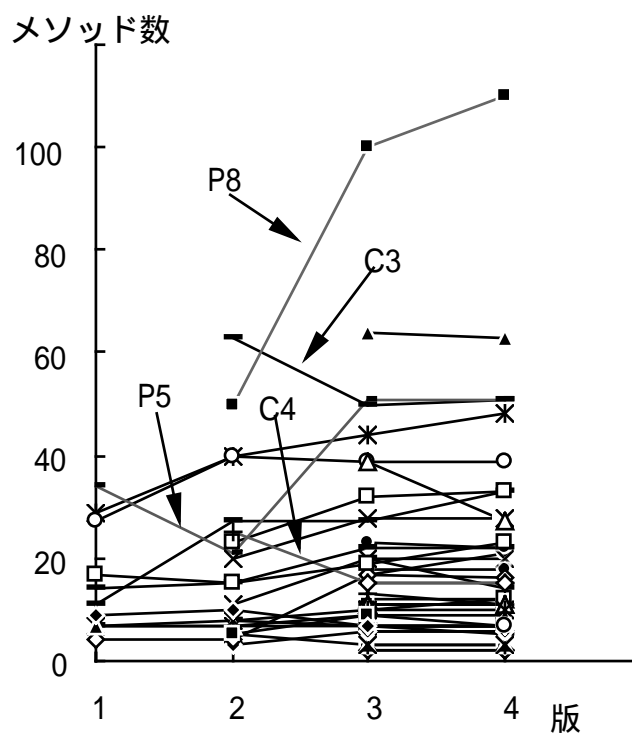


図 2.6: SystemA のクラスのメソッド数から観た進化過程

変更を行った結果，最大の規模を持っていたクラス P5 は他のクラスと同程度の規模となった．

2. 第 2 群のオブジェクトの進化：継承構造の再構成

第 2 群に属するクラスには，第 2 版で，最大の規模を持つ計算クラス木に属するクラス C3 が含まれている．図 2.7 に，SystemA の第 3 群で観測された継承構造の再構成過程を示し，各クラスのメソッド数を括弧内に示した．SystemA のクラス C3 は第 2 版から第 3 版に至る過程でメソッド数を 63 から 50 に減らし，この間に，クラス C3 のスーパークラスは，第 2 版でスーパークラスを共有していたクラス C2 へ変更された．クラス C2 は第 2 版でクラス C4 ともスーパークラスを共有していたが，第 3 版ではクラス C4 のスーパークラスとなった．図 2.6 に見られるように，クラス C4 の進化過程でも，第 2 版から第 3 版に至る過程でメソッド数を 25 から 15 に減らすといったクラス C3 と同様の变化を観測することができる．第 2 版から第 3 版に至る過程の継承構造の変更は，これら 3 クラスに着目した設計変更であり，クラス C3，C4 の規模を縮小させる効果をもたらしていた．

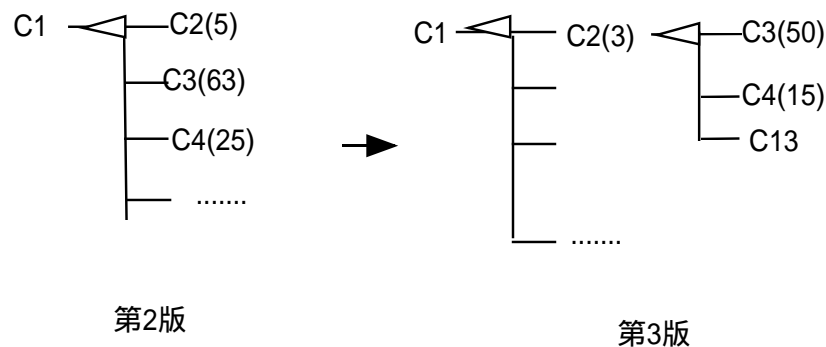


図 2.7: SystemA における継承構造の再構築（クラス名とそのメソッド数（括弧内）， \triangleleft は継承を表す）

第 1 群，および第 2 群で設計変更対象となっていたクラスは，いずれも設計変更時の版の最大メソッド数を持っていたクラスであったことから，開発者が規模の大きいクラスを設計変更対象として注目していたと解釈

できる。ただし、第 3 版で最大規模となっていたクラス P8 に対する設計変更が行われなかったのは、利用者環境の変化が収束しており、再設計を行うコストと、再設計によって向上する拡張性の改善との均衡が成り立たなくなったためと考えられる。

3. 名称変更

SystemA では第 2 版から第 3 版に至る過程で、9 クラスがその名称を変更している。これは第 3 版の全クラスの 1/4 に該当する。名称変更は、熱交換シミュレーションの設備の多様化によって、新たに追加された類似設備に対応するクラスと既存クラスとを区別するために行われていた。漸進型開発では、利用者環境の変化によって徐々に開発者の問題領域への理解が深まり、ここで観測したように、1/4 のクラスの名称が変更される場合もある。この例から、開発者にとって、そのクラスの役割の識別子として使われているクラス名称は、開発の全期間で変化しないとは言えないことが確認できた。

2.3.2 SystemB の進化

SystemB では、表 2.2 に示したように、最大規模を持つメソッドは安定していた。これは、規模が大きいことが必ずしも設計上の問題を持ったオブジェクトであるとは限らない例である。したがって、設計上の問題点は、ある時点における規模だけではなく、オブジェクトの進化過程から発見しなければならないと言える。

SystemB では、SystemA で行われたようなシステムのフレームワークを変更する設計変更は起きていない。しかし、図 2.8 に示すクラス継承木の構造の変化では、クラスの統合、クラスの削除、継承木の再構成、継承木の消滅といったオブジェクトの技術的環境への適応を観測できた。

1. クラス統合

SystemB の第 1 版では、入金消し込みの顧客に関するデータは、顧客の認証を行う画面操作を行うためのオブジェクト、顧客を認証するために必要な情報を保持するオブジェクト、その他の顧客登録情報を保持する

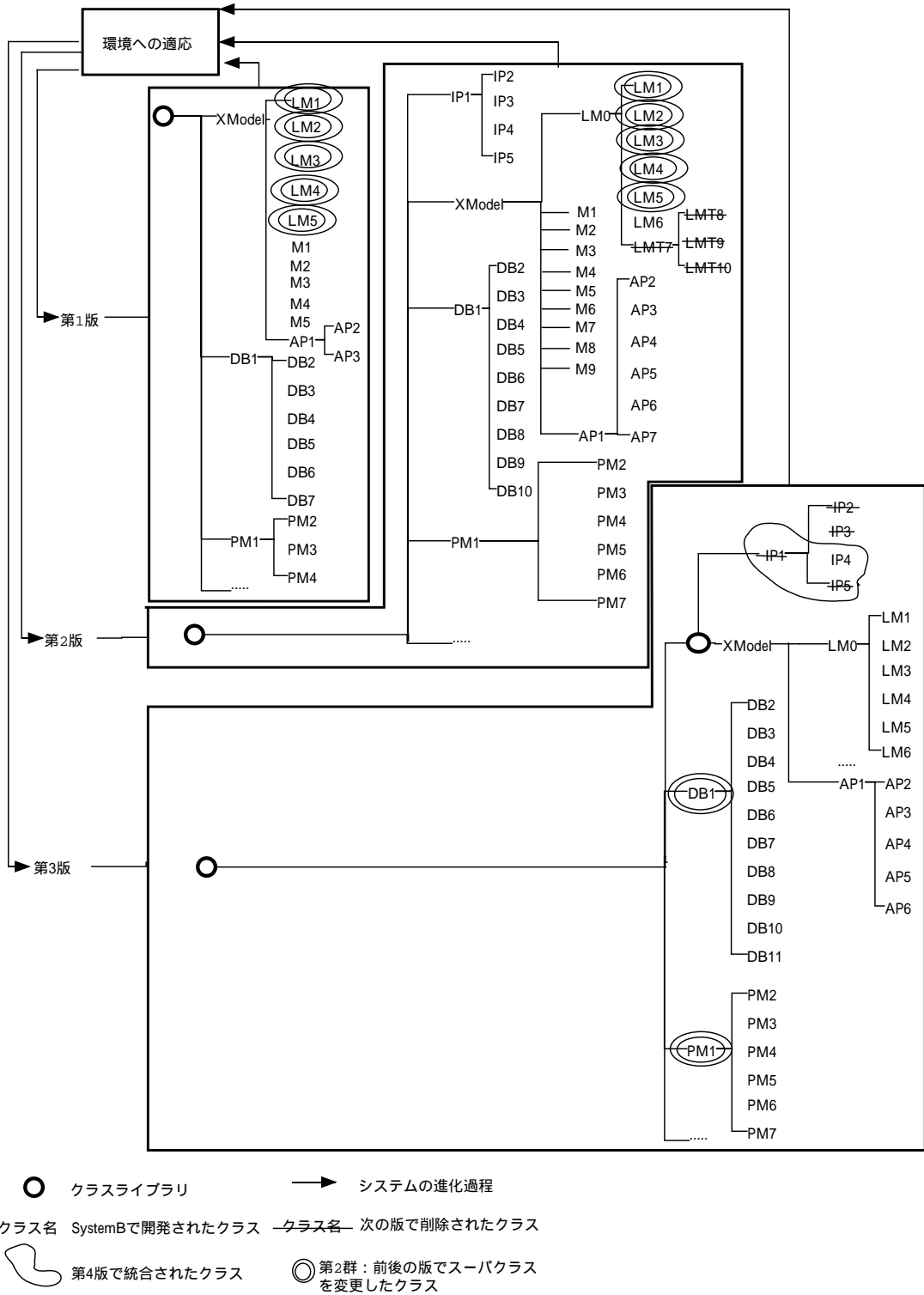


図 2.8: SystemB のクラス継承木の進化過程

オブジェクトの 3 つに分割されて管理されていた。しかし、第 2 版では、これらのクラスは統合されて新たなクラスが定義された。この設計変更は、クラスの凝集度を向上させる効果がある。

2. クラス削除

表 2.7 は SystemB におけるクラス削除に関する開発者へのアンケート調査結果を示した表である。この表のクラス分割の項目には、クラス分割によって名前変更が行われたものも含まれている。表から、不要によるクラスの削除は全体の 25% にすぎず、クラスの分割 / 統合によるクラス削除が 75% を占めていることがわかる。

名前が変更された 13 クラスについて、名称変更理由を調査したところ、次の項目が挙げられた。

- 日本語名称を英語名称に変更
- 類似クラスの混乱をさけるため、より詳細な仕様を表現できるクラス名称に変更
- より適切な名前を発見。

第一の原因は非英語圏の開発者に発生する技術的環境の変化である。第二と第三の理由は、SystemA で観測したクラス名称変更理由と同様であり、開発者が問題領域を理解するという、クラスを取り巻く技術的環境の変化によるクラスの進化と解釈できる。

表 2.7: クラス削除の理由とその割合(()内はメソッド数を表す)

理由	削除されたクラス数	割合 (%)
統合	9	20.9
分割	10	23.3
名称変更	13	30.2
不要	11	25.6
合計	43	100.0

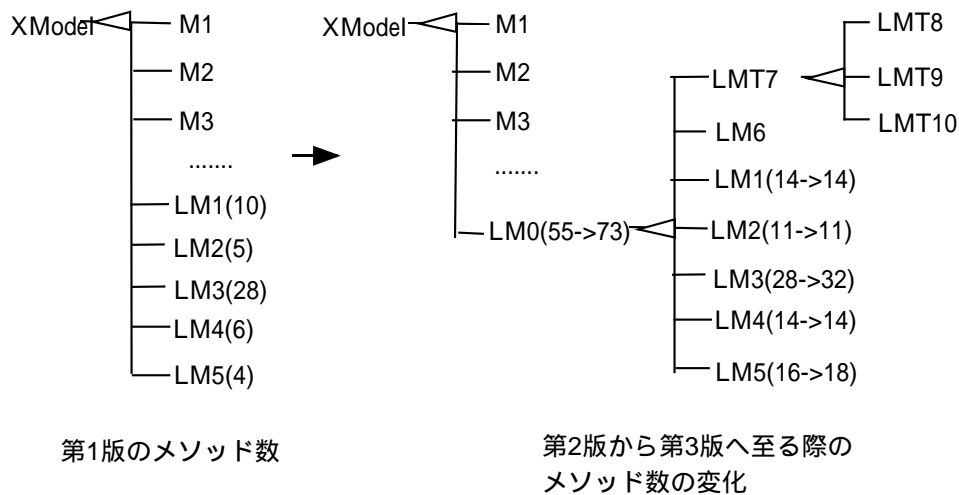


図 2.9: SystemB における継承構造の再構築

3. 継承木の再構築

図 2.9には、SystemB における継承構造の再構築の例を示す。第 1 版のクラスの横に表示した数字は第 1 版のクラスのメソッド数を表している。第 2 版と第 3 版のメソッド数は矢印の前後に示した。これらのクラスは図 2.8に示したように、SystemA で観測されたクラス進化の第 2 群に属するクラスである。クラス XModel は第 1 版で 11 のサブクラスを持っており、第 2 版では、新しいクラス LM0 を継承するクラスとして、このうちの 5 クラスが選択されて継承構造の再構築が行われた。これらのクラスは、一覧表で表示された入金情報や顧客情報の多重選択や選択対象に対する情報の絞り込みといった操作に関する役割を分担するクラスであり、利用者インタフェースの追加要求を反映させるための設計変更の対象となったクラスである。

設計変更の方針にはいくつかの選択肢が考えられるが、ここでは、新たな機能が要求されたクラス群に対して新しいスーパークラスを定義し、複数のクラスに追加すべき機能をこのスーパークラスに吸収させ、サブクラスの進化を緩やかにする効果を得ている。第 2 版から第 3 版に至る期間は、利用者インタフェースの追加要求とともに、システムの最終版

の仕様が決定され開発された期間である。この期間に、クラス LM0 は、そのメソッド数を 55 から 73 に増加させているが、クラス LM0 のサブクラスのメソッド数は、クラス LM0 ほど増加していない。

4. 継承木の消滅

図 2.8 のクラス LMT の継承木は第 2 版から第 3 版の期間で消滅した。この開発期間では、構築したシステムと別会社で開発されたデータベースとの結合が行われており、消滅した LMT のクラス継承木はデータベースが提供されていなかった第 2 版でデータベースを擬似的にテストするためのクラスであった。このような継承木の消滅は、システムの開発時に予想されるものであり、予測できない環境変化に適応する進化というよりもシステムの成長過程の 1 事象と捉えられる。

継承木の消滅は、その他に、直線的な継承構造を避けるための設計変更でも起こる。漸進的な開発が進められる過程では、一度定義した継承構造がクラスの削除などにより直線的な継承構造や無意味な継承構造となってしまうこともある。図 2.8 のクラス IP1 の継承木では、第 4 版の継承木消滅の過程で、残されたサブクラスとスーパークラスとの統合が同時に起きている。このクラス継承木に属する 5 クラスのうち、再利用したクラスが IP1, IP2, IP3, IP4 の 4 クラスあったが、第 4 版を開発する時点で、再利用クラスが提供していた機能を他のクラスのメソッドとして取り込むことによって、IP2 と IP3 の 2 クラスが不要となった。また、残された SystemB のために開発されたクラス IP5 の機能は、継承木内に残っていた再利用クラス IP4 に取り込まれ統合された。その結果、これらのスーパークラスであったクラス IP1 の抽象化の意味がなくなり、この継承木は消滅した。ここでは、再利用クラスの見直し、再利用クラスの進化、継承構造の見直しによる再利用クラスの再構成が行われていたことになる。いずれの設計変更も利用者へ提供する機能は変化しないため、技術的環境への適応と考えることができる。

機能追加は利用者環境の変化である。しかし、新しい利用者環境にクラスが適応しようとしたとき、クラス構造やシステムの構造が複雑になり、可読性や将来の生産性が低下することはある。そこで、技術的環境に適応する必要が生

じる．設計変更は，このような問題を解決するために経験的に選択されてきたが [56]，その妥当性は，ここで調査したように，設計変更後のクラスの進化過程によって評価することができる．

2.4 考察

Lorenz らは，平均値から大きく外れた値は不適切な設計であることを表わしていると言っているが [37]，本研究によって，メソッドの規模だけから設計上の問題を指摘できないことが明らかとなった．たとえば，複雑な計算を行うメソッドの行数は，メソッドの平均行数に比べて大きくなるが，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傾向を続けて進化することである．規模が大きく，変更が多いということは，バグを生む原因も高まると考えられる．したがって，オブジェクトの変化を時系列で観測し，進化過程の特徴から特異な進化を示すオブジェクトを発見することには意味がある．

進化過程の調査から設計上の問題点を発見できるのは，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一般的に観測された傾向を根拠にしている．平均値や中央値が大きく変化しないオブジェクトは，メソッド行数の他に，クラスのメソッド数，行数がある．付録 B に，各システムのメソッド行数，クラスのメソッド数，クラスの行数の各頻度分布と，幾何分布より求めた理論的な分布とを重ねたグラフを掲載した．いずれの分布も，理論値から求めた頻度分布によく従う．頻度分布の特徴から，将来開発されるクラスやメソッドの規模と数を見積もることが可能であると考えられる．しかし，頻度分布を見積もりに応用するためには，さらに他のシステムについて調査を行う必要があるだろう．

