

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

Quantitative Analysis on Evolution Process of Object-Oriented Systems

中谷多哉子* 玉井哲雄†,
東京大学大学院広域システム科学

友枝敦‡ 酒匂寛§
(株)SRA, SI ビジネス事業部

概要

オブジェクト指向によるシステム開発では、クラスライブラリの再利用やコンポーネントの活用によって、短期間でシステムを開発できるようになってきた。また、このようなシステムの開発では、ユーザ要求の進化への継続的な適合を意図した漸進的开发が取り入れられている。オブジェクト指向の場合は、この漸進的开发の中に、将来の拡張のための再設計も含まれることが多い。

我々は、システムの初期開発と保守のプロセス、さらに新システムへの移行、そしてクラスライブラリやフレームワークを構築するための再設計プロセスを含めて、システムの進化プロセスと呼ぶ。本稿では、システムの進化プロセスを計測し定量的な分析を行うことで、その特徴を理解すると共に、開発者の再設計を支援する情報の提示が可能であることを述べる。

1 はじめに

従来から、進化型プロトタイプを採用した漸進的开发を行えば、ユーザ要求の変更に対応でき、さらに長期に対応し続けることができるという利点が指摘されてきた [2]。漸進的开发の場合、新規開発プロセスと保守プロセスとを区別するよりは、むしろ両者を合わせてシステムの進化プロセスと捉えた方がよいだろう。また、オブジェクト指向のシステム開発では、システムの進化に従って徐々にそのアプリケーション、あるいはドメインのフレームワークが理解され、確立されていくため、漸進的开发が適ると言われている。

ところが、このようなシステムの進化プロセスを定量的に計測し、そこから開発上の示唆を得ようという研究はあまり行われていない。時系列に基づく定量的なシステムの計測の例として、その計測が人員の再配置やシステムの再構築を指示する管理者の意思決定支援に役立ったという報告がある [6]。しかし、この研究で計測されたのは実績工数と進捗であり、見積りやスケジュールとの差分が時系列でどのように変化するか

を示すためのものであった。これは、設計内容を定量化し、開発者の意思決定を支援するというものではない。

そこで我々は、オブジェクト指向システムの設計内容を定量化し、進化プロセスに従った時系列の計測値を収集することで、開発者の意思決定を支援する可能性を探った。本稿では、システムの進化プロセスの中で、特にクラスの進化プロセスに着目し、計測のためのメトリクスを実際のシステム開発に適用した計測結果を示すとともに、その分析結果を述べる。

2 進化メトリクス

システムの進化プロセスを定量化するにあたり、観測レベルとしてシステム、クラス、メッセージ、メソッドの4つを対象にメトリクスを定義した。これを進化メトリクスと呼ぶ [5]。その中で、クラスの設計内容を定量化するために用いたメトリクスは次の通りである。

- *CLOC*: コメントを含む行数
- *NIV*: 定義されたインスタンス変数の数
- *CNOM*: 定義されたメソッドの数
- *DIT*: 最上位のクラスから数えた継承の深さ (または *ADIT*: 着目しているクラスから見て一番近いライブラリクラスまで継承の深さ)

3 計測実験と結果

3.1 概要

計測実験に用いた温度調節シミュレーションシステムは、VisualSmalltalkの一部を構成する PARTS Workbench 上に構築された。このシステムは、分析から最終的な納品までの8ヶ月を一人の開発者が担当し、その間、6回に渡って開発中のシステムを顧客に提示した。開発者は、このシステムが顧客に提示される度に要求変更を受け取り、次の開発に反映させた。したがって、これらの版は、ひとつのシステムの進化プロセスを代表するものとみなしてよいだろう。

*tina@graco.c.u-tokyo.ac.jp

†tamai@komaba.ecc.u-tokyo.ac.jp

‡tomoeda@sra.co.jp

§sakoh@sra.co.jp

システムの最終的な基本構成は、次のようになった。

- 表示部品：シミュレーション対象の設備を視覚的に組み立てて、シミュレーションの結果を表示する部分
- 編集部品：シミュレーション用のデータを入力する部分
- 計算部品：シミュレーションを実際に計算する部分

表 1: システムの進化の概要

<i>metric</i>	<i>ver.1</i>	<i>ver.2</i>	<i>ver.3</i>	<i>ver.4</i>
クラス数	16	26	47	52
行数	1560	3714	8044	8677
メソッド定義数	193	436	885	927
クラス木の数	5	6	6	6

ver.0 から ver.1 までは 2 ヵ月、それ以降の版はそれぞれ 1 ヵ月毎に顧客へ配布された。進化の検討は、使い捨て型のプロトタイプであった ver.0 と、その計測結果が ver.4 とほとんど違わなかった ver.5 を除き、ver.1 から ver.4 までを対象とした。

表 1 でクラス木の数が ver.1 から ver.2 にかけて増加したのは、ver.1 の表示部品からシミュレーションの計算部分だけを抜き出し、ver.2 の計算部品を定義したためである。

3.2 計測結果と定量的分析

システムの進化に従った時系列の計測結果を基に、クラスの進化の特徴を分析した。

1. 「*CLOC*, *NIV*, *CNOM* の計測結果の分布は、進化プロセスを通してあまり変化しない。」

CNOM の分布の状態を表 2 に示す。この分布は右側へずそを引く尖った分布で、システムの進化に従ってこの傾向は顕著になっていく。しかし、中央値は ver.2 の例外を除いて 12 前後で安定している。*CLOC* と *NIV* の計測値の分布も同様の傾向があった。

2. 「全クラスの約半数は計測値の変化量が小さい。」

CNOM を基にした、クラス毎の計測値の変化の様子を図 1 に示す。横軸はシステムの版を表し、縦軸はその版で計測された値を表す。図からいくつかのクラスで計測値の増減が観察できるが、システムの進化の前後でどの位変化したかを求めるために、増減率を次の式で定義する。

$$\Delta CNOM_{c_i} = \frac{CNOM_{c_{i+1}} - CNOM_{c_i}}{CNOM_{c_i}}$$

表 2: *CNOM* の分布

	<i>ver.1</i>	<i>ver.2</i>	<i>ver.3</i>	<i>ver.4</i>
平均	15.1	19.4	19.6	18.3
中央値	11	15	12.5	11
標準偏差	10.3	16.5	19.4	19.9
尖度	-0.7	1.1	6.4	9.21
歪度	0.9	1.3	2.3	2.66
最小	4	3	2	2
最大	34	63	100	110
データ数	11	21	42	47

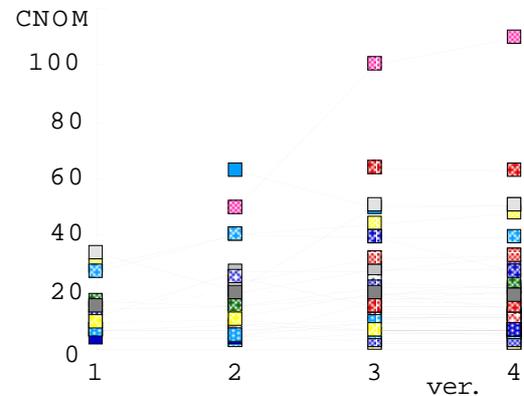


図 1: クラスの進化プロセス

ここで、 $CNOM_{c_i}$ は、クラス C の $ver.i$ における *CNOM* の計測値を表し、 $\Delta CNOM_{c_i}$ は $ver.i$ から $ver.i+1$ までの間の増減率を表す。

この増減率を各クラスのそれぞれの進化の前後で計算し、その絶対値の最大値を抽出して分布を集計した。その結果を表 3 に示す。

表 3 から、最大の増減率が 2 割未満のクラスは全体の 49.0% を占めていることがわかる。

表 3: *CNOM* の最大変化率の分布

増減率 (%)	クラス数	割合 (%)
0 ~	23	49.0
20.0 ~	14	29.8
40.0 ~	5	10.6
60.0 ~	5	10.6
合計	47	100.0

3. 「抽象化が行われ当該クラスのスーパークラスが新たに定義された場合や、兄弟クラスが追加され

スーパークラスの再設計が行われた場合、*CNOM*の値を減少させることがある。」

スーパークラスが新たに定義されると、その仕様を継承できるようになるため、当該クラスの*CNOM*の値は減少すると予想される。この例は、表示部品及び計算部品で観測できた。これを group.1 として図 2 に表示す。

また当該クラスの兄弟クラスが追加されるとスーパークラスの設計内容が変更され、その結果、*CNOM*の値が減少することもあると予想される。計測結果では、この例を ver.2 から ver.3 にかけて進化したクラスで観測した。これを図中に group.2 として示す。この現象が起きた期間では、同時にスーパークラスの*CNOM*の値の増加も観測できた。

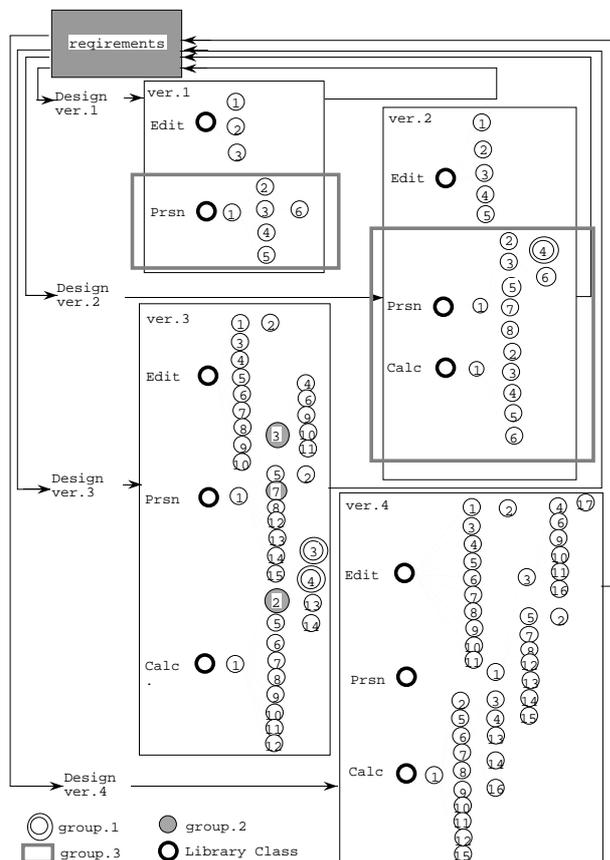


図 2: クラス木の進化 (Prsn:表示部品, Edit:編集部品, Calc:計算部品)

- 「システムを移行させずに複数のクラス木に影響が及ぶ再設計は、進化の初期の段階であれば起こりやすい。」

調査を行ったシステムは、表示部品、編集部品、計算部品の 3 クラスが一つの単位になって、シミュレーションを構成する設備を実現していた。この

3 クラスで構成されるシステムの構造は ver.2 で確立されたものである。ver.2 に至る進化の部分をも、図 2 に group.3 として示した。

今回の計測実験では、開発者へのインタビューも行った。インタビューの結果、ver.3 の開発を始めるとき、表示部品と計算部品の再構築を検討したが、変更による影響範囲が大きく、利用者へ配布する期間を満足できないことが予想されたため、取り止めていたことがわかった。

したがってシステムの広範囲に及ぶ再設計は、進化の初期に行われても、システムの規模が大きくなるにしたがって行われにくくなると考えられる。

- 「*CLOC*, *CNOM*, *NIV*には強い相関があり、例えば *CLOC/CNOM*の値は、クラス木毎に固有の値を持ち、進化を通して一定である。」

計測した結果から、相関分析を行った結果、これらの値には強い正の相関関係があることがわかった。

例として、編集部品のすべての版を横軸に *CNOM* を、たて軸に *CLOC*をとった散布図にプロットした(図 3)。図から明らかなように、編集部品の *CLOC/CNOM*の値はほぼ一定で、4 版の平均は 19.4 であった。同様の傾向は他のクラス木にも観察でき、表示部品の値は 8.2、計算部品の値は 6.0 であった。

- 「あるクラス木の平均的な *CLOC/CNOM* 値から大きく外れた値を持ったクラスは、再設計の対象候補と考えることができる。」

図 3 で 2 点を結んだ矢印は、ver.3 で初めて定義されたクラスの、ver.3 から ver.4 までの変化の様子を表す。ver.3 の計測値は他の同じクラス木に属するクラスとは異なる値を示したが、ver.4 の計測値は平均的な値になっていた。これは、開発者が再設計対象として抽出したクラスが、平均的な *CLOC/CNOM* 値を示していなかったクラスであり、再設計の結果、そのクラスが平均的な値を持つように修正されていたことを意味している。

したがって、計測値の分布図が設計上の例外的な部分を識別し、設計を再考する手がかりにできることが示唆される。

4 考察

4.1 進化メトリクスの有効性

Chidamber らや Lorenz らによって、クラスの複雑度や可読性を定量化するためのメトリクス [1] や、見積りのためのメトリクス [4] が提案されている。興味深いことに、我々の計測値をシステム全体で平均した値は、Lorenz らが示したシステムの平均の行数や 1 クラス当たりのメソッド数などと非常に近い値を示した。しか

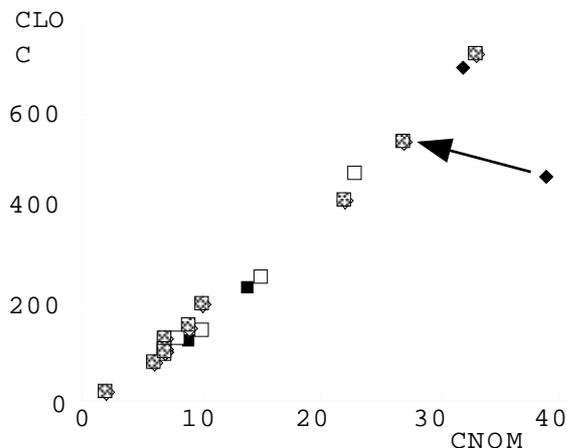


図 3: CLOCとCNOMの散布図 (編集部品)

し、設計の評価は、システム全体の平均値を基にして議論するよりも、クラス毎の固有の値を基に行った方が、設計者の直感とよく合うことを確認した。

4.2 システムの進化プロセス

どのような場合に計測値が減少するかについて検討を行ったが、全てのクラスで兄弟クラスが追加されたときに計測値が小さくなったわけではない。仕様の追加によって、減少分が相殺された場合や、クラス固有の手続きを追加するために再定義が必要な場合があったものと思われる。いずれにしても、このような再設計の作業をクラス木の範囲内で行えるのであれば、設計の初期段階で頑健な継承構造を定義する必要はないかもしれない。

ところで、クラスレベルでは進化の過程で再設計が行われるため、プログラムが進化し続けると複雑になるという法則 [3] は、必ずしも当てはまらない。再設計のうち、複数のクラス木を対象にしたものは、開発の初期に一度行われただけで、それ以降は行われていなかった点には注意が必要だろう。このことから、システムの全体構造を決めるフレームワークが安定したものであり、要求の進化に追従して進化しにくいものであると結論できる。そこで、あえてシステムの寿命を短く設定し、システムの要求規模にあった、より良い構造を順次開発する形態を取り入れることには意義がある [7]。

今後は、このシステムの再構築後の計測を継続して行い、移行の前後でシステムのフレームワークがどのように変化するかを明らかにしていきたい。

5 まとめ

本稿では、システムの進化プロセスに従って進化するクラスの様子を定量的に分析することで、クラスの特徴を表す進化メトリクスの計測値が進化によって変化しない分布を持つこと、 $CLOC/CNOM$ の値はクラス木毎に固有であること、また開発の過程で、抽象クラスなどのクラス木内の再設計が行われるといった進化の特徴を明らかにした。

また、定量的な分析によって、再設計対象の構成要素を抽出でき、再設計後の目標値を示せる可能性を述べた。

謝辞

本研究は、通商産業省ならびに情報処理振興事業協会の推進する独創的情報技術育成事業の一環として行われたものである。研究の機会を与えてくださった同事業関係者の皆様に感謝いたします。

参考文献

- [1] Chidamber, S. R., & Kemerer, C. F.: "A Metrics Suite for Object Oriented Design," *IEEE Transactions on Software Engineering*, Vol.20,No.6, pp. 476-493 (1994).
- [2] Davis, A. M.: "A Strategy for Comparing Alternative Software Development Life Cycle Models," *IEEE Transaction on Software Engineering*, Vol. 14, No. 10 (1988), pp. 1453-1461.
- [3] Lehman, M. M.: "Programs, Life Cycles and Laws of Software Evolution," *Proc. IEEE Spec. Issue on Software Engineering*, Vol. 68, No. 9 (1980), pp. 1060-1076.
- [4] Lorenz, M. & Kidd, J.: *J. Object-Oriented Software Metrics*, Prentice Hall, 1994.
- [5] 中谷, 友枝, 酒匂, 玉井: "オブジェクト指向によるシステムの進化メトリクスの検討," ソフトウェアシンポジウム'96, SEA, 1996, pp. 52-62.
- [6] Stark, G., Durst, R. C. & Vowell, C. W.: "Using Metrics in Management Decision Making," *IEEE Computer*, September, 1994, pp. 42-48.
- [7] Tamai, T. & Torimitsu, Y.: "Software Lifetime and its Evolution Process over Generations," *Proc. of the Conference on Software Maintenance*, November, 1992, pp.63-69.