

第 6 章

オブジェクトの組織化と進化モデル（仮説）の洗練化と一般化

事前調査で得られた進化の傾向から，システム，クラス，メッセージ名，メソッドの進化モデルを構築できるが，本研究では，クラスの進化モデルの一般化を中心に検討を行った．図 6.1は，1995 年度に定義したオブジェクトの進化モデルを，今年度の調査に基づいて一般化したクラスの進化モデルである．

進化モデルは，再利用対象のライブラリクラスとアプリケーション固有のアプリケーションクラスから構成される．クラスの進化過程は，次の 5 種類へ分類される．第一の進化過程は，まだ仕様が固まらず小さな進化を繰り返すが，進化の規模は全体の分布の内堀を越えない程度である過程，第二の進化過程は盛んに設計者の手が入り仕様が大きく変化する過程，第三の進化過程はほとんど設計者の手が入ることがなく，したがって進化が起きない過程であり，第四の進化過程はライブラリクラスとして再利用可能なクラスとなる過程で，進化をほとんど起こさない過程である．最後の第 0 進化過程にあるクラスのオブジェクトは，たとえプログラム上に存在していたとしても他のオブジェクトとメッセージの送受信を行わない過程で，この過程にあるクラスは，全く進化しない．

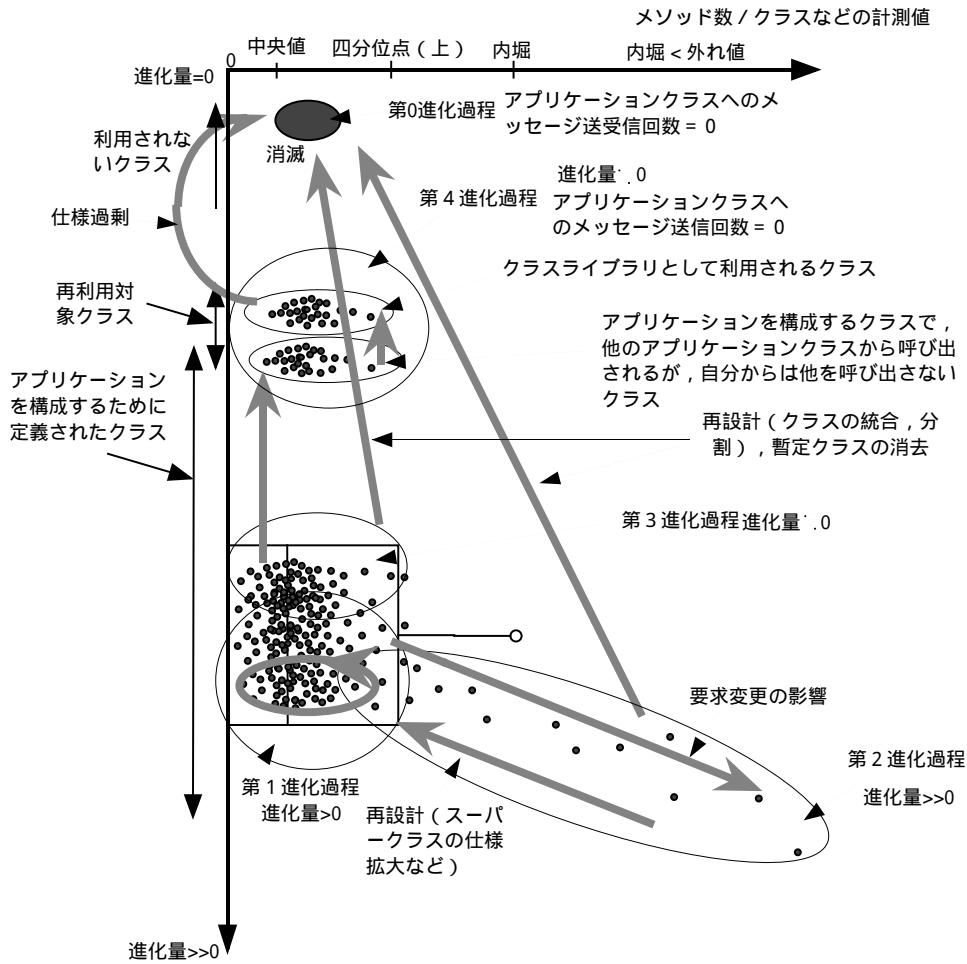
図 6.1に示した進化モデルの進化過程は，昨年度および今年度の調査結果で観測されたクラスの進化過程を対応づけて定義した．

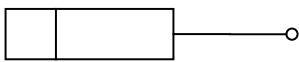
- 第一進化過程

3つのシステムにおいて，クラスの *CNOM*，*CLOC* といった値の分布で，内堀以内の観測値を持つほとんどのクラスを，第一進化過程へ分類する．第 4章に示した図 4.5，4.6，4.7に観られるように，平均値付近に分布するクラスは，そのメソッド数をあまり変化させていない．そこで，第一進化過程の定義として「クラスの進化の量がシステムの進化の量に依存せず，小さく，進化後も内堀から出ることはないクラス」とする．

- 第二進化過程

第二進化過程に分類したクラスは，熱交換シミュレーションシステムと入金消し込みシステムから抽出した．熱交換シミュレーションシステムでは，システムの進化と共に規模を拡大する少数のクラスを観察した．入金消し込みシステムでも多くの役割を持つクラスに，熱交換シミュレーションシステム程顕著ではなかったが，規模の大きい進化を観測した．入金消し込みシステムの開発者へのアンケート調査で説明したように，入金消し込みシステムに定義されているクラスのうち，変化が大きかったクラスの変更理由の第一は設計変更によるものであった．図 4.6でも，図 4.5でも，いくつかのクラスが規模を縮小させているが，規模を縮小させたクラスでは，いずれのシステムでもスーパークラスへの仕様の抽象化のような再設計が行われていた．



註) 

計測された値は、右に尾を引く尖った分布になる。
 この絵は、計測値から求めた箱髭図である。箱の左側は負の値を表わすが、進化メトリクスには負の値は存在しない。箱の中線は中央値と表わし箱の大きさは標本の上下の四分位点で区切られた標本の境界を表わす。右に伸びた髭は内堀で止めた。内堀の値は、上下の四分位点の差を1.5倍し、その値を上側の四分位点に足した値。外れ値とは、内堀の値よりも大きな値をもった標本の値を言う。

図中の●はオブジェクトを表わす。
 ➡ オブジェクトの進化過程

第1進化過程：小さな進化が起きる過程
 第2進化過程：大きな進化が起きる過程
 第3進化過程：進化がほぼ止まっている過程
 第4進化過程：ライブラリとして汎用化と仕様の洗練が行われる進化の過程
 第0進化過程：今後進化が起きることがない過程

図 6.1: クラスの進化モデル

以上から、第二進化過程の第一の特徴として、Belady & Lehman の進化法則の「進化するプログラムは、構造の劣化を反映させる複雑度を増加させないような保守を行わない限り、増加する。」を対応させる。

これらの2システムが利用者の要求変更によって進化したのに対し、証券管理システムは、利用者からの要求変更といった開発組織の外部からの刺激なしに開発されたシステムである。このシステムが成長する過程を組織化過程としたが、図 4.7では、組織ができあがったと思われる時期以降、いずれのクラスも進化していないようである。また、他の2システムでも、ver.3以降の要求変更の量は少なく、再設計も起きていない。以上のことから、第二進化過程のもう一つの特徴として、「要求変更による組織の変更が起きない限り、クラスの再設計は行われず、クラスの進化も起きない」を挙げる。

- 第三進化過程

2システムの ver.1 から ver.3 に至る要求変更が大きかった時期に、進化が起きていないクラスがあった。また、動的メトリクスを用いた計測結果から、メッセージの送信回数と受信回数が共に他のクラスに比べて少ないクラスは、進化も他のクラスに比べて小さいことがわかった。メッセージの送受信回数が少ないクラスは他のクラスとの結合度が小さいので、他のクラスの変更の影響を受けることが少ない。

このように、要求変更の影響や他のクラスの進化の影響を受けずに進化しないクラスを第三進化過程に分類する。第一進化過程に定義されたクラスは、小さな進化を繰り返し、やがて要求変更に対応できる役割を持つようになる。このような状態に達したクラスは、他のクラスとの結合度が小さい限り、他のクラスの変更の影響を受けることもなく、安定した状態を保つことができる。このような状態に到達すると、クラス階層上の再設計が行われなくなり進化する必要はなくなる。これが第三進化過程である。クラスの適切な役割分担が行われていれば、第一進化過程にいたクラスは、システムの進化に伴って第三進化過程へ向かって進化する。

- 第四進化過程

ライブラリクラスは第四進化過程に分類される。ライブラリクラスがアプリケーション開発の中で定義される場合、他のアプリケーションクラスとメッセージ送受信を行う通常のクラスとは異なり、自分から他のクラスにメッセージを送ることはない。クラスをこのようなライブラリクラスの条件にあうように設計することで、システムの進化の状況に依存せず、一定の役割を提供し続ける安定したクラスにできる。最初に第一進化過程に定義されたクラスも、他のクラスとの結合をなくすことによって、第四進化過程のライブラリ的クラスにできる。

また、第四進化過程にあるクラスは、完全なライブラリクラスまでには進化の余地があるクラスと、進化がほぼ止まったライブラリクラスへ分類できる。

入金消し込みシステムでライブラリ用クラスとして定義されたクラスのうち、2クラスについては、ほぼ同時期に開発が進行していた証券管理システムでも再利用されていた。これらのクラスは日付に関するクラスで、それぞれのシステムで独自に進化していた。この例に見られるように、進化の余地があるクラスは、品質が保証されているわけではなく、個人の間で再利用され仕様が変更されることもある。Smalltalk のライブラリクラスの場合は、全く変更されずに再利用されたり、継承を定義して再利用されることが再利用の方法として推奨されている。個人の間で再利用される第四進化過程にあるクラスが、Smalltalk のライブラリクラスのようなクラスになるためには、利用実績による進化が必要である。

- クラスの進化過程の変遷

第一進化過程から第四進化過程までの各進化過程は、この順に進化するというわけではない。クラスが各進化過程をどのように変遷するかを図 6.1に矢印で示した。

新しいクラスは、第一進化過程に生成される。第一進化過程に属するクラスは、進化が進み、安定期を向かえて第三進化過程へ進化するクラスと、次第に役割を増し、構造を複雑化させて第二進化過程へ進化するクラス、他のクラスとの関連を切り第四進化過程へ進化するクラスとがある。第四進化過程へ進化するクラスは、アプリケーション内でライブラリ用クラスとして取り扱われるクラスで、第四進化過程へ進化すると他のシステムで新たな進化過程を迎えることもできるようになる。

また、第四進化過程で、再利用の実績を持ったクラスは、仕様内容が開発者の間で既知のものとなり、共通のライブラリとして取り扱われるようになる。この状態になると内部の構造や規模に変化が起きなくなる。

第二進化過程へ進化したクラスは、規模を縮小させるような再設計、すなわちスーパークラスへの役割と構造の抽象化やクラスの分割が行われない限り、規模の拡大や複雑化の傾向を持つ。だから、第二進化過程のクラスが再設計を受けると、第一進化過程へ戻る場合もある。

第0進化過程への移行は、すべての進化過程から起こりうる。

以上の各過程でクラスが進化するとき不変なのは、そのクラスが属するクラス木毎に固有な1メソッド当たりの行数の値である。

これらの進化過程では、フレームワークを構成するクラスの進化過程や、オブジェクト指向プログラミング言語の環境で提供されているクラスライブラリの進化過程を議論していない。この議論は今後の課題である。