

オブジェクトの進化プロセス

玉井 哲雄[†]

オブジェクト指向システムの進化プロセスを考える。オブジェクト指向に代表される現在の、またこれからのソフトウェアプロセスにおいては、開発と保守の境界が不明になるだけでなく、開発環境と運用環境の境界も不明になると考えられる。その意味では全体を進化プロセスとして捉えることが、ますます重要となる。

On Object Evolution Process

TETSUO TAMAI[†]

1. はじめに

オブジェクト指向技術はようやく浸透しつつあり、オブジェクト指向分析や設計の方法論も実践の段階にある。さらに実用面からの期待が高い再利用についても、個々のオブジェクトをクラスライブラリから検索して利用するという段階から、より大きな構造単位での利用という段階に議論が進んできている。すなわち、設計パターンや、応用分野ごとの基本的な枠組み (framework) という概念が注目を浴び、実際に多様なパターンや枠組みが提案され利用され始めている。ある種の一般性の高い枠組みは、コンポーネントウェアという製品として提供され始めてもいる。

これまでのオブジェクト指向技術が想定していたのは、主として新規に開発するシステムへの適用であった。再利用にしても、それを新たな開発にどう生かすかという視点で考えられている。しかし、社会にはすでに膨大な遺産ソフトウェアが存在する。遺産 (legacy) という聞こえはいいが、実のところは老朽化したソフトウェアというべきものである。現在の産業界における大きな課題のひとつは、このような老朽化した巨大ソフトウェアをいかに保守しまた蘇生させるかである。情報機構の基盤構造が、小型化 (down-sizing)、分散化、開放化に向かって変化するにつれ、老朽化ソフトウェアを蘇生させるなら、そのような新しい基盤に移行した上で再構築するという要請が出るのも自然である。しかし、ただでさえ難しい再構築作業に、新た

な基盤上での設計という技術課題が上乗せされるという問題もある。

ここでももちろん、オブジェクト指向技術に期待がかかる。その際、忘れてならないのは、オブジェクト指向技術に基づいて再生が成功したとしても、その後システムはさらに変化していくという点である。そこで問題は、遺産ソフトウェアをオブジェクト指向技術にいかに移行させるかというものと、移行後のシステムをいかに進化させるかというものと、両者ということになる。その意味で、オブジェクト指向システムの進化プロセスを把握することは重要であるが、これまでのところこの分野を対象とした研究はほとんどなされてこなかったように思われる。

そもそもオブジェクトは動的に変化する性質を持つ。もっとも基本的なレベルでは、オブジェクトが互いにメッセージをやりとりしながら自らの状態を変化させる。また、オブジェクトの生成、消滅という事象は、従来の手続型プログラムに自然には備わっていない概念である。さらに、より長期の時間軸で考えた場合に重要なのは、オブジェクトの内部構造や、オブジェクトによって構成される組織の成長・進化というプロセスである。

われわれは、オブジェクトの作る組織とその変化に注目し、これを生命システムや社会組織との類比から組織化と進化プロセスとしてモデル化することを目標とする。すなわち、

- オブジェクトの組織化：各オブジェクトがどのような単位を作り、それらがどのような組織を構成するのか。

[†] 東京大学総合文化研究科

Graduate School of Arts and Sciences, University of Tokyo

- オブジェクトの進化: 各オブジェクトがどのように進化し、それらが構成する組織がどのように変化していくのか。

を分析し、その枠組みとなるようなモデルを構築したい。

2. 二つの視点

生命や社会との類比は分析の枠組みとして有効であるが、そのような類推のみに頼るのは、方法論として脆弱である。この研究を進める際の視点として、分析的な立場と構成的な立場との二つが考えられ、とくに前者に対して生物などとのアナロジーが役立つが、同時に後者の立場を軽視しては技術としての内容が乏しいものとなるだろう。

分析的な立場とは、対象を自律系システムとそれが環境との相互作用を行う場としてとらえ、それを観察・分析することで、組織化および進化プロセスの一般的なメカニズムや法則を発見しようとするものである。一方、構成的な立場とは、対象は自然物ではなく人工物であり、組織化や進化といってもそれを引き起こすのはあくまでも人間であるという認識を出発点とする。したがって発想は工学的であり、目的は優れた設計論を構築するところにある。ここでは、分析的な立場と異なり、設計としての善し悪しという価値判断が入る。ただし、この場合の設計とは、一つのシステムに対して一回限り行われる作業と考えるのではなく、システムの成長・進化に伴って持続する長期のプロセスとする。

3. 保守と進化プロセス

ソフトウェアの保守という作業を進化という視点で見直そうという考え方は、Belady & Lehman に遡る¹⁾。しかし、この視点は次の2つの傾向により、ますます重要になってきたように思われる。

開発と保守の境界の喪失

実際、開発プロジェクトといっても、まったく未経験でゼロからすべてを作るというケースは、非常に少ない。また、保守作業も(変更)要求の分析定義、設計、プログラミング、テストと開発と同様の工程をたどる。すなわち、通常の開発プロセスに既存システムの理解という作業がさらに加わったものが保守プロセスと見ることもできる。さらに、長期に渡るソフトウェア開発では、開発の途中でユーザの要求変更や設計変更がしばしば起こる。これは開発工程の中で、要求仕様や設計仕様の保守作業が発生しているのと同様である。

逆に、継続的なソフトウェアの保守による進化プロセスは、開発が引き延ばされて続いている状態と見ることもできる。

これらに関連し、ソフトウェアのライフサイクルモデルも、スパイラルモデルのような進化型のもの、さらには再利用過程を組み入れるもの、といった形に発展してきている。

開発/保守環境と実働環境の境界の喪失

システムとして反応型のものが中心となると、システムは常に動作をしていることに意味があり、計算結果を出して停止するというものではなく、とくに分散環境でそのようなシステムが動作している場合、それに「保守」を加え、また新たな機能を付け加えることも、運用と並行して行なわざるをえなくなる。コンポーネントウェアのような開発環境は、その意味では運用環境と同化しつつあるといえよう。

4. アプローチ

現在までの研究方法として、このテーマに実証的に取り組むため、具体的なシステムを事例としてとりあげて、その進化プロセスをなるべく定量的に測定し、その結果を分析することから、プロセスにおける普遍的な特性、法則を探るといった形を取っている。これまでに対象とした事例としては、

- 熱処理シミュレーション・システム
- 証券管理システム
- ネットワークサービス顧客管理システム

などがある。その結果については、以下の参考文献^{2)~4)}に譲る。

参 考 文 献

- 1) Belady, L. A. and Lehman, M. M.: A Model of Large Program Development, *IBM Systems Journal*, Vol. 15, No. 3 (1976), pp. 225-252.
- 2) Nakatani, T., Tamai, T., Tomoeda, A. and Sakoh, H.: Quantitative Analysis on Evolution Process of Object-Oriented Systems, *Proceedings of the International Symposium on Future Software Technology*, October 1996, Xi'an, China, pp. 49 - 56.
- 3) 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, 1996年9月, 筑波, pp. 389-392.
- 4) 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, 1996年5月, 広島, pp. 52-62.