

オブジェクト指向、UML に関する教育の視点と実践

中尾 信明

株式会社富士通九州システムエンジニアリング

オブジェクト指向によるシステム開発を目指して、企業内ではオブジェクト指向、UMLに関する教育に対するニーズが高まっている。

本論文では、オブジェクト指向、UMLに関するこれまでの企業教育の経験に基づき、効果的な教育の視点と実践方法を述べる。また、オブジェクト指向やUMLを用いた分析、設計技術の習得のために、教育をどのように行えば効果的であるかを述べる。

The Point and Practice of Education concerning Object Oriented and UML

NAKAO Nobuaki

FUJITSU KYUSHU SYSTEM ENGINEERING LIMITED

In this thesis, an effective education aspect and method of practice are described based on the experience of a current enterprise education concerning object-oriented and UML.

1. はじめに

昨今のシステム開発を取り巻く環境は、非常に厳しいものとなっている。これまでのレガシーを中心としたアプリケーションの開発から、Web アプリケーション、分散アプリケーションへの開発へ移り、高い技術を用いながらも、短納期、高品質を求められるようになった。

そのため、これまでの手続き型のシステム設計から、オブジェクト指向による設計が重要視されている。また、急な要求にも柔軟に対応し、かつ生産性の向上を謳ったプロジェクト管理論や開発プロセス論が議論されている。

このような背景を元に、多くの企業では開発における分析工程から構築工程までの基本となる考え方であるオブジェクト指向と UML を開発現場へ急速に導入を進めようとしている。よって、これらに関する企業内教育や開発プロセスの導入教育のニーズが高まってきている。(図 1 参照)

本稿では、これまでの講義形態での企業内教育の実践から、オブジェクト指向、UML の技術的な視点や、オブジェクト指向開発プロセス、プロジェクト管理などのマネジメントに対する考え方を、受講生へ基礎から短期間で効果的な教育を実施するために、強調すべき点、留意する点を述べる。

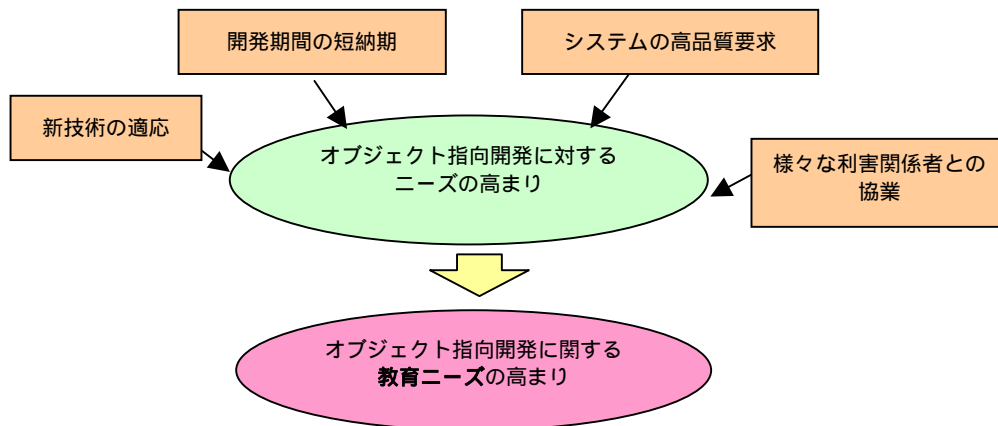


図 1 オブジェクト指向開発に関する教育ニーズの高まり

2. オブジェクト指向の理解のために

2.1 従来の開発方法論とオブジェクト指向の明確化

受講者の多くは、これまで行ってきた開発において、どのような考え方に基づいて分析や設計が行われてきたのかを意識することなく開発を進めてきた場合が多く、無意識のうちに手続き型指向に陥っている場合がある。このため講義の冒頭で、これまでの開発指向としては、手続き型指向が多く用いられてきており、これまでの受講者の開発においても、この考え方に基づいた開発であったということを認識させ、受講者が採用している開発思考を意識させる必要がある。

次に、これからオブジェクト指向に関する知識や実際の分析、設計を学ぶ目的を明確にするために、オブジェクト指向と従来の考え方の比較を行うことによりメリット、デメリットを挙げて、オブジェクト指向の優位性を訴える。これらのメリットやデメリットをトータル的に考えて、オブジェクト指向に対する期待と広がりの可能性を受講者に理解させ、これからの講義に対しての目的意識をはっきりさせることが重要である。

2.2 現実世界とオブジェクト指向との関係の明確化

オブジェクト指向を理解させるためにシステムの世界との関わりを最初から強調するよりも、身の回りの物事をオブジェクト指向で考えられるようになることが、ここでの目標となる。

オブジェクト指向に対する初学者やシステム開発未経験の受講者は、一般的に「オブジェクト指向」は難しい考え方とと思っていることが多い。このため、最初の導入部分として、我々が生活し

ているこの社会や我々が社会を認識する方法がオブジェクト指向と密接に関わっていることを強調することにより、オブジェクト指向に関する意識の障壁を早い段階から取り除くことが重要である。ある意味、自然な考え方であるオブジェクト指向を、システムの開発に持ち込んだ考え方、またこの考え方を体系的にまとめたものがオブジェクト指向の性質であると強調することにより、受講者はシステム開発とオブジェクト指向との関連性を認識することができる。

2.3 オブジェクト指向とシステム開発とのマッピング

受講者に、オブジェクト指向がこの世の中の現象をモデリングしたものであると理解させたとしても、すぐにはオブジェクト指向とシステムの開発とを関連付けさせたり、利用させたりすることは困難である。

この段階では、現実世界の関連をモデリングするため考え方としてのオブジェクト指向から、システム開発のために分析、設計を行うためオブジェクト指向へ転換させるために、さらに踏み込んだ考え方を提示していく。つまり、現実社会に存在する形のあるものや関係を、オブジェクト指向を元にモデリングしていく単なるモデリングツールとしてのオブジェクト指向から、モデリングされたクラス群から、最適な設計となるようなシステム的设计に変換するためにオブジェクト指向を用いるといった考え方への変換が必要となってくる。この最適な設計のための性質としては、継承、カプセル化、インタフェースなどが挙げられる。つまり、オブジェクト指向を用いて現実のものや状況を表現することを第一段階の理解とすれば、オブジェクト指向を理解した上で、システム的设计に適用させることが次の段階の理解と言える。また同時に、一つのクラス(オブジェクト)をシステムの1実装単位として実装するといった実際のシステムの実装上におけるクラスと、オブジェクト指向での物を表現したクラスとの関連を明確にしておくことも重要である。

単なるオブジェクト指向の用語を理解しただけに留まってしまえば、どのようにシステム開発や開発言語と関連があるのかといった根本的な理解が得られないため、本来の目的である「システム開発のためのオブジェクト指向」を学ぶためにはこの過程は非常に重要である(参照)。

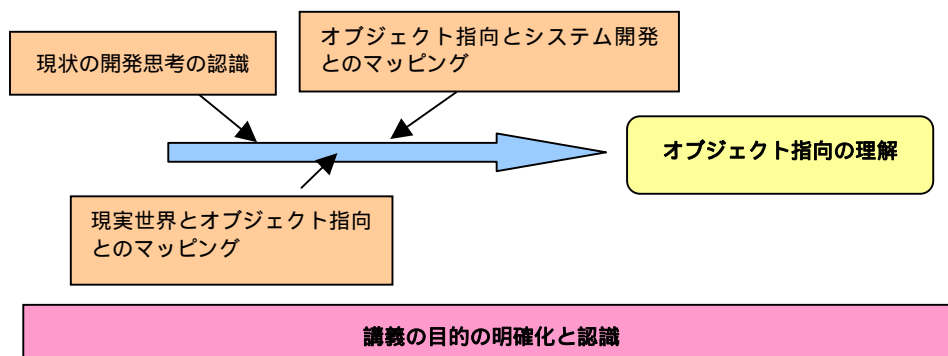


図 2 オブジェクト指向理解のための要素

オブジェクト指向とシステム開発との結びつきを説明する例として、Java などのソースコードを用いたクラスの説明が有効である。なお、プログラム言語を全く経験していない受講者に対して

も容易なソースコードを用いての解説は行うべきである。Java に代表されるプログラム言語はその性格上、非常に直感的にクラスを表現できるため、システム開発の初学者に対しても理解の一助となる。

3. UML の理解のために

3.1 モデリングの意味の理解

UML とは Unified Modeling Language の略称である。これはオブジェクト指向によって表現される様々なもの、関係などを記号化して表現するための表現方法である。

まず、UML とオブジェクト指向との関連を明確に受講者に対して伝達すべきである。オブジェクト指向によって表現できる性質を図式化し、共通認識を与えるために導入され、システム開発の利害関係者で共有できるなどのメリットを強調する。また、開発の工程に応じた UML を使い分ける必要があることを留意しなければならないことも受講者に対して認識させる必要がある。

また、モデリングによる効果を明確に認識させることが大切である。

3.2 自由度の高い UML の理解

受講者が UML に関して、「UML はどこまで厳密に表記法を規定しているのか。」という点を強く意識しがちである。この点においては実際に UML は必要最低限のルールしか規定していないことを強調すべきである。「UML の表記表によって表現された情報では、説明が不足してしまう。」という問いや、「UML では表現できない情報を追加したい場合はどのようにすべきか。」ということで受講者が混乱する場合がある。

この場合は「メモ」といった方法で補足することができるなどの柔軟な対処方法が存在していることを理解させなければならない。重要なことは、UML が開発者の思考の表現を制約するものではなく、思考を表現するために定められた最低限のルールであるという認識を持たせることが重要である。

3.3 UML は開発のための道具であるという認識の明確化

UML の表記法を講義する上で、もっとも陥りやすいポイントは UML で様々な関係や振る舞いを表現することに重点を置いてしまい、その結果 UML を図示することによって表現することで満足してしまう点である。

もちろん、UML を用いてモデリングを行えるといった能力は重要であるが、システム開発において UML で表現すべきものは、システムを開発するために分析、設計した結果であり、分析、設計する能力が重要であることを認識させなければならない。受講者に対しては、UML に関する一定の理解が成されたところで、UML はシステム開発のための分析、設計を行うためのツールであるという視点を再確認、再認識させると、今後の UML を用いた分析、設計段階に、スムーズに移行できる。

3.4 UML とシステム開発との関連とを意識

実際のシステム開発において、企業ごとに様々な開発ドキュメントが用意されている。UML による表記だけでは、これまでのドキュメントに網羅されている開発に必要な情報をすべて記述することはできない場合がある。

しかし UML を用いて分析や設計を行うためには、これまでの開発ドキュメントに記述された情報を UML に変換しなおすことの重要性やメリットを強調しなければならない。その際には、ここまでに学んだ知識であるオブジェクト指向、UML、そしてこれらを利用するターゲットとなるシステム開発との3つの関連を意識させ続けることが重要である。

4. オブジェクト指向開発の理解のために

4.1 パターン適用による分析、設計技術の取得

システムの開発において、すでに存在しているものを機械的に UML を用いて表現することは求められていない。むしろ、漠然とした現象の中から要求を見つけ出し、その要求を満たすために必要なオブジェクトを発見していくといった作業が重要である。オブジェクト指向分析、設計では要求を明確にし、明確になった要求の一つ一つを着目し、その要求を満たすために必要なオブジェクトを抽出する必要がある。また、要件から出てくるクラス以外に、システムとして機能を実現させるためのクラスを登場させる必要がある

また、分析や設計は経験や知識を必要とするため、初学者に対して教えることは非常に困難である。また、受講者が思い思いに設計を行う結果、最終的にオブジェクト指向が求める分析・設計となり得ない可能性が多く、分析や設計が様々なパターンとなってしまおうといった弊害が生まれる。

これらの弊害をなるべく排除するためにこの「パターン指向」を導入する。有名なパターンの一つとして「MVC モデル」が挙げられる。MVC モデルの詳細な説明は行わないが、手続き型指向からオブジェクト指向を学んだばかりの受講者に、この考え方を利用してシステムの分析、設計を経験させる場合、スムーズにオブジェクト指向の考え方や設計方法に移行させることができる。また、パターンを利用することで見通しの良いオブジェクトの選別や関係の抽出ができることを理解させることができる。

4.2 UML による受講者の思考を表現

オブジェクト指向や UML を用いた分析・設計手法を体験するために、実際に、機能要求書から UML を利用したドキュメントを作成するといった演習を行うことが効果的である。

この演習では、講師の方から機能要求書の詳細な説明は行わない。なぜならば機能要求書に対する読解力を養うとともに、受講者個人の解釈（他人と違って全く構わない）を奨励しているためである。

演習では、できるだけ実際の開発現場における雰囲気や条件を経験させるかということが重要である。

4.3 アウトプットとしての UML を説明

UML を受講生に記述させる演習のあとに、解答例を講師の方から受講者全員に対し、一方的に解説を行い、その解答例を受講生の記述した UML との違いを指摘する方法をとっていた。しかし、この一方的な講義スタイルでは個々人の理解度を把握できないばかりか、適切な指摘ができない可能性がある。

そこで、数人を 1 グループとしてグループで一つのアウトプットを出すように指示をする。こうすることによって、アウトプットをまとめるといった作業を通して各人が自分の考えを、UML をベースに他のグループ構成員に説明する機会を作ることとなる。グループ作業によるメリットは様々なものが挙げられる。(表 1 参照)。これらの作業を通して、UML は開発者同士の共通言語であることを再認識させるねらいがある。

グループ作業によるメリット
<ul style="list-style-type: none">・ グループメンバーとの議論が必要となり、分析・設計に関する理解を深めることができる。・ グループ内の討議のため、UML の記述を行う必要があり、記述スキルが付く。・ 議論を行うことで、分析・設計能力を向上するのに役に立つ。・ 議論を通して、グループでの話し合いの進め方を体験することができる。

表 1 グループ作業によるメリット

4.4 他人の設計との差異を理解する重要性

先週の最終的なアウトプットとして、グループで一つまとめることを一つの目標とさせる。このアウトプットをグループでの作業を経て得られた構成員の統一見解をグループごとに全受講者に向かって説明をしてもらう。グループ内で一つの結果を導き出すといった作業を通して、グループ内で様々な考え方があることを認識してもらう。ここで大事なことは、どのような分析、設計結果に対しても肯定的なコメントを出すことである。前にも書いた通り、分析や設計という作業は、これが絶対的に正しいといったものがあるわけではない。クラスの責務が明確に分類されているか。メッセージを送るタイミングは間違っていないか。などの基本的な誤りは指摘すべきであるが、独特な責務を持ったクラスを導入して処理を完結させようとするそのアイデアを評価すべきである。

確かに、これまでの講義の中でこちらが予想をしていなかった分析、設計を行う受講者もいるが、この作業を通してもっとも大切なことは、自分たちの考えを矛盾なく UML に置き換える技術だと考える。そして、その上でパターンを最大限利用した最適なモデルを構築するための助言を与えるべきである。

また議論を通して他者の考えを、UML を通して理解することができ、なぜこのような設計をしたのかといったような深い議論に発展することができる。ここまでの過程により、UML での表現、

設計手法の議論等、比較的レベルの高い視点での問題提起が行うことができる。

4.5 実プロジェクトへの適応のために

講義の一つである「オブジェクト指向開発」の中では、開発プロセスも同時に講義を行っている。この開発プロセスは RUP (Rational Unified Process) に代表されるようなインクリメンタル手法をベースとしたものである。

昨今、短納期でも品質が高く、仕様の変更にも柔軟に対応できる開発方法論が様々な形で議論されている。

現実には、受講生がこれまでに関わってきたプロジェクトでは、ウォーターフォール型が一般的であり、教育を受講したからといってすぐさま開発方法を変更することは困難である。

よって、この講義の目的としては昨今の開発方法論の議論が活発に行われていることと、一つの先進的な開発方法論のモデルを紹介しながら、プロジェクトを行う上での留意点を意識させることが重要である。受講者は1人の実装者としての視点のみを持っている傾向が強いが、講義を通して実装者という立場とともにプロジェクト管理者としての視点も同時に意識させることができる。

また、分析や設計などの演習を通してオブジェクト指向を用いた開発を受講者に経験させる。実プロジェクトでは、演習のように比較的容易に行うことは難しく、モデリングを行うことは多くの経験や知識を必要とするであろう。しかし、オブジェクト指向の初学者にとってこれらの講義や演習での作業は、これからのモデリング作業で注目する点やヒントを与えることになるだろう。

5. 結論

オブジェクト指向言語の広がりにより、システムの開発にオブジェクト指向を導入の検討や、期待が高まってきているため、オブジェクト指向や UML などの技術を短期間で習得する教育にニーズが高まっている。

それぞれの企業にとって、開発方法論は様々である。しかし、これからのシステムの開発では、様々な利害が絡んだ人間同士の関わりが多くなり、意思疎通の手段が求められることとなる。このようなニーズからも、システム開発において UML による表現が広がっていくと思われる。

この状況の中で、オブジェクト指向を利用した分析、設計の出来る人材を早く、効果的に育成することは非常に重要なことである。

受講者の理解を確実にするためには、オブジェクト指向の基本をしっかりと理解させることが重要である。また、オブジェクト指向がシステム開発における一つの視点であり、UML で表現すること自体が目的ではないことを十分に理解・認識させることが重要である。

また、オブジェクト指向を用いた分析、設計方法を理解させるためには、パターンを提示することで、オブジェクト指向を用いた演習を行うことが効果的である。この演習では、個々人で作業を行うのではなく、数人を1グループとして作業を行い、その中で受講者一人一人が他人との議論を行うことで、自然とオブジェクト指向の考え方で UML を用いて設計をすることができるようになる。

また、「開発方法論」というものが存在し、様々な議論がなされていることを早い段階から意識さ

せることで、実装者の視点とともにプロジェクト管理者の視点も育てることとなる。
受講生の中でも、実際のプロジェクトにどのように適応していくかが大きな課題であるということ
を、これからの問題点として認識している。
実際の開発プロジェクトに生かすためには、講義で培った知識と経験の上で、自分たちの会社の
文化や開発形態に応じた新しい開発方法を導いていく必要がある。そうでなければ、オブジェク
ト指向とは単に教育のための限られた思考としかならないだろう。

6. 参考文献

- 1 . UML によるオブジェクト指向開発実践ガイド
吉田裕之、山本里枝子、上原忠弘、田中達雄 技術評論社
- 2 . オブジェクト指向入門 富士通ラーニングメディア
- 3 . UML 入門～表記法の取得～ 富士通ラーニングメディア
- 4 . UML によるオブジェクト指向設計演習～実践的開発技法とプロジェクト管理～
富士通ラーニングメディア
- 5 . オブジェクト指向によるシステム設計・開発 情報処理振興事業協会
富士通ラーニングメディア

“Java” は、米国 Sun Microsystems, Inc. の米国およびその他の国における商標または登録商標です。

“Rational Unified Process”, “RUP” は Rational Software Corporation の商標または登録商標です。

“UML” “Unified Modeling Language” は Object Management Group Inc. の商標です。