オブジェクトとロール: 二元論モデルの考察

玉 井 哲 雄†

オブジェクトを構成要素とする一元論的モデルに対して，オブジェクトとロールに基づく二元論的モデルが有効な状況が多く見られる．ここではそのような 2 元論モデルをサーベイしそれらの特徴を明確にするとともに，関連するわれわれの研究を紹介する．

Objects and Roles: A Dualistic Model

Tetsuo Tamai†

1. 双対概念としてのオブジェクトとロール

ソフトウェア工学の歴史の中で，数多くのモデルやモデル構築方法論が作られてきた[7]．たとえば，データフロー・モデル，状態遷移モデル，ER モデルなどが挙げられる．モデルを構成するのは，その要素となる元とそれらの間で成り立つ関係ないし構造である．したがって，要素として何を選び，要素間の関係として何に注目するかがモデルの本質を定める．

オブジェクト指向技術の普及に伴い，モデルの要素としてオブジェクトを取ることが一般化した．オブジェクトはデータ的性質とプロセス的性質とを合わせ持つのみならず，分析，設計，実装を通して同じオブジェクトという概念により一貫したモデルの構造を維持できるという利点を持つ．

オブジェクトを基本とするモデルは，ある意味で一元論的な世界を形成するといえる．これに対して，オブジェクトと直交するロールという概念を導入し，二元論的なモデルを作る方が適切な場合がある．以下では，そのようなモデルの特徴を議論する．

2. 協調モデル

多くの OO 方法論では，オブジェクトによる協調に重点がおかれる．協調の記述方法として典型的なものに，使用事例 (use-case) がある．使用事例は I. Jacobson により提唱され，現在の統合モデル化プロセス (Unified Modeling Process)[2]でも継承されている．UML では協調を記述するのに，使用事例図とともに，事象系列図や協調図を用いる．

† 東京大学大学院総合文化研究科
Graduate School of Arts and Sciences, University of Tokyo

Wirf-Brock 等は協調を記述する際に，オブジェクトがその協調で果たすべき「責任」という概念に焦点を当てる[8]．そのためにクラスと責任と強調の関係を表にした CRC カードを使うことを推めている．

これらの方法論では，役割 (role) という言い方は使われていないが，それに相当する概念を協調に参加するオブジェクトの持つべき性質の一面として捉えている．実際上は，協調や責任を並べ挙げることで必要なオブジェクト集合やその性質を同定することが主目的とされる．そこでロールに相当する概念は，ほぼメソッドに対応し，オブジェクトとロールは必ずしも対等の関係にない．

OO 方法論の中には，ロールにより重要な位置づけを与えているものがある．たとえば OOram 方法論[4]が典型的な例である．そこでは協調を記述するためにロール・モデルを構築すること，さらにロール・モデルを合成して統合モデルを作ることがステップとして組み込まれている．また，D. Riehle はロール・モデルをさらに進めて，設計パターンの合成やフレームワークの構築に利用することを提案している[5]．

分析や設計段階で現れるロールは，プログラミング段階では消えてしまうのが普通である．しかし，プログラムにも明示的にロールを位置づけられるようにしようとする試みも，行なわれている．プログラミング言語にそのような構成要素を持ち込む方法もなされているが[9]，既存のプログラミング言語を使いながらロールを具体化している例に，VanHilst & Notkin[6]がある．彼らは C++のクラス・テンプレートを用いてロールを記述し，それを再利用することを提案している．

3. 適応，発展，移動

協調におけるロールの役割に見られるように，ロールとオブジェクトの間には，オブジェクトは複数のロールを果たすことができ，またロールは複数のオブジェクトによって演じられうるというように，相補的な関係が存在する．このスキームはより広い目的に適用可能である．とくに，オブジェクトに適応，発展，移動といった性質を与えるのに適した枠組になっている．

3.1 適　　応

本田等は Morphe という言語に「適応化コンポジション」という概念を導入するにあたり，次のような例を問題意識の出発点として挙げている[11]．「花子は太郎と結婚し，家庭という環境に適応して振舞う．その後，花子はある研究所に入り，研究者として適切に振舞うように適応する．」

このようにオブジェクトが新しい環境に適応したり，今までいた環境が変化したためにそれに適応するという状況を実現するために，さまざまな試みがなされている．たとえば分散 AI やマルチエージェントの分野では，自律的に環境に適応するエージェントの開発が中心課題の一つとなっている．中島等の Gaea はその代表的な試みといえよう[10]．しかし，技術としてはまだ未開拓な部分が大きく残されたアプローチといえよう．

オブジェクトとロールの 2 元モデルは，この環境適応の目標に対して，簡潔ながら十分に強力な枠組を提供する．本田等の方法では，たとえば家庭という環境のオブジェクトとしての記述の中に，夫や妻という属性を持つ部分オブジェクトの記述があり，それと花子というオブジェクトが結合する際に，花子は妻の属性を獲得するような変換が行なわれるという手法をとっている．ここではオブジェクトの双対概念としてロールを導入するという考え方はとられていないが，環境を構成する部分オブジェクトはロールに相当するものと考えてよい．

鵜林&玉井の Epsilon モデルでは，協調の場としての環境を，ロールの集合とそれらの間の相互動作として明示的に記述する[9]．オブジェクトはロールのインスタンスと結合することにより，環境に適応する．この結合の際には，Morphe のようなオブジェクトの変換は行なわず，状態同士は直積がとられ，互いのメソッドを上書きないし名前換えすることで，相互作用を実現する．この方法だと，オブジェクトがいくつかのロールと結合したり，一旦結合したロールと分離したりする機構が簡潔になるとともに，分離後にもオブジェクトとロールそれぞれが自己の状態を持続することが可能となる．

3.2 発展と移動

適応の見方を変えると，オブジェクトの進化・発展や移動という概念に結びつく．環境に時間的に適応していく過程を発展と捉えることができ，空間的な環境の変化に適応する過程を移動と捉えることができるからである．たとえば Gottlob 等はデータベースのスキームの時間的な変化という発想から，ロール概念を導入している[1]．また，粂野等の Flage という言語では，移動エージェントを実現するのに環境への入出という概念を用いている[3]．ただし，Flage ではロールという概念はなく，オブジェクトが環境に入るとすべて同じ機能を獲得するようになっている．

参 考 文 献

1) G. Gottlob, M. Schrefl, and Röck. Extending object-oriented systems with roles. *ACM Transactions on Information Systems*, 14(3):268–296, July 1996.
2) I. Jacobson, G. Booch, and J. Rumbaugh. *The Unified Software Development Process*. Addison-Wesley, Reading, 1999.
3) F. Kumeno, H. Sato, T. Kato, and S. Honiden. Flage: A programming language for adaptive software. In *Proceedings of ICSE'98*, volume 2, pages 103–108, 1998.
4) T.Reenskaug, P.Wold, and O.Lehne. *Working with Objects: the OOram Software Engineering Method*. Manning Publications, Greenwich, 1996.
5) D. Riehle and T. Gross. Role model based framework design and integration. In *OOPSLA '98*, pages 117–133, Vancouver, Oct. 1998.
6) M. VanHilst and D. Notkin. Using Role Components to Implement Collaboration-Based Designs. In *OOPSLA '96*, pages 359–369, 1996.
7) R. Wieringa. A survey of structured and object-oriented software specification methods and techniques. *ACM Computing Surveys*, 30(4):459–527, 1998.
8) R. Wirfs-Brock, B. Wilkerson, and L. Wiener. *Designing Object-Oriented Software*. Prentice Hall, Englewood Cliffs, 1990.
9) 鵜林尚靖，玉井哲雄．動的な役割変化を考慮したオブジェクト間の協調動作記述とそのモジュール化メカニズム．電子情報通信学会論文誌, J82-D-I(6):718–729, 1999.
10) 中島秀之，野田五十樹，半田剣一．有機的プログラミング言語 gaea．コンピュータソフトウェア, 15(6):503–516, 1998.
11) 本田康晃，渡滋，所真理雄．適応化コンポジション–開放型システムにおけるコンポジションに基づいた新しいソフトウェア構築手法．コンピュータソフトウェア, 9(2):122–136, 1992.