

オブジェクト協調と役割モデル

玉井 哲雄[†]

オブジェクト指向分析で役割モデルを活用することが注目されているが、本論ではオブジェクトの協調の場を記述するために役割モデルを根本的に考え直し、ソフトウェアの基本構造を構築するためのモデルとなる可能性を探る。

A Role Model for Object Collaboration

TETSUO TAMAI[†]

Role models have been attracting certain interests among OO analysts. In this paper, we reconsider characteristics of role models for writing collaboration schemes and pursue the effective fundamental model of software architectures.

1. はじめに

オブジェクト指向技術は成熟しつつあり、実践的な普及が進んでいる。しかし現代の複雑で多様なソフトウェア・ニーズに応えるには、複数の視点から実世界をモデル化する方法論、システムの動的振舞いについての洞察を伴う設計、広い範囲の再利用を可能とする手法が求められる。そのためには、従来のオブジェクト指向分析/設計で行われてきた、オブジェクトのみを構成要素とする 1 元論的なモデルではなく、オブジェクトと役割 (role) という双対的な概念に基づく 2 元論的な計算モデルを作ることが有効であることは、これまでも論じてきた^{4),5)}。

本論では、とくにオブジェクトの協調の場を記述するという目的から、改めて役割モデルに要求される基本的な性質を考察し、ソフトウェア基本構造モデルとなりうるための条件を明らかにする。

2. 協調モデル

われわれの提案する役割モデルの意義は、次の 3 点に要約される。

(1) [関心事の分離]

主体指向プログラミング (subject oriented programming)²⁾ や局面指向プログラミング (aspect oriented programming)³⁾ では、関心事を分離し、複数の視点や設計の方針が混じりあ

て事態が複雑になることを避ける工夫を提唱している。われわれの提案する役割モデルでは、1 つの関心事を 1 組の役割による 1 つの協調場に対応させることにより、関心事の分離を自然に実現する。

(2) [オブジェクトの機能の多様性と適応性]

あるオブジェクトが同時にいくつかの役割を兼ねたり、動的に役割を変えたりすることは、従来のオブジェクト指向の枠組では実現が困難であった。たとえば M. Fowler はそのための種々の工夫を提案している¹⁾。役割にオブジェクトと独立な存在として市民権を与え、オブジェクトが役割と結びつく機構を実現することで、オブジェクトが同時に複数の役割を演じたり、動的にある役割と結合したり分離したりすることが、可能となる。

(3) [部品化]

再利用の単位として、オブジェクトの他に役割が独立した部品となりうるのみならず、役割が協調する場そのものが部品化/再利用の対象となりうる。しかも、役割や協調の場を言語としてサポートすることにより、それらが実行可能な部品単位となり、設計パターンの直接利用が可能となりうる。

次に、これらの基本要件を実現するために必要となる構成要素を考える。

3. モデルの構成要素

基本的に、次のような要素および機構を記述する仕組が必要である。

[†] 東京大学大学院情報学環

Interfaculty Initiative in Information Studies
Graduate School of the University of Tokyo

- (1) [場と役割を定義し生成する仕組]
場も役割も、オブジェクトと同様に状態と動作を持つものとして定義でき、また動的に特定の場と其中的役割を生成できる必要がある。これは場と役割をオブジェクトと同様にクラスとして定義できるようにすればよい。
- (2) [場に役割をカプセル化する仕組]
協調の場に役割をカプセル化し、役割が互いのみを参照して相互作用する記述を可能とする。基本的には Java の内部クラスと類似の仕組があればよい。しかし、協調動作はインスタンス間で実行されるべき性質のものであるから、インスタンスとしての役割を簡潔に記述できることが望ましい。同時に、役割のクラスから動的に不定数のインスタンスを生成する仕組も必要となる。
- (3) [オブジェクトと役割を結合しまた分離する仕組]
オブジェクトは一般に複数の場の役割と動的に結合でき、また結合を解くこともできるようにすればよい。この結合は動的であることを要求することから、オブジェクトのインスタンスと役割のインスタンスが結合するものとする。結合に実際的な意味を与えるには、結合したオブジェクトと役割の状態間や動作間に何らかの相互作用/干渉が起ることを想定しなければならない(互いに独立に動作するのみなら、結合する意味がない)。そのためには、役割側から見てオブジェクトに要求すべきインタフェースを記述できればよい。オブジェクト側から見た役割のインタフェースは、あらかじめ既知として記述することにしても不都合ではない。なぜなら、協調の場が大きな部品化/再利用の単位だからである。

4. 今後の研究方針

以上の基本的な枠組みのもとで、これまでに提案してきた Epsilon モデルを見直し再構築すること、そしてその上でなるべく具体的で実際的な例題を多く記述することを、当面の方針とする。

例題としては、これまでも Epsilon モデルの例題として取り上げてきた契約ネット・プロトコル、SOP や AOP で取り上げられている代表的なもの(たとえばプログラム解析木の上で多様な機能を実現する問題)、設計パターンとして提案されている種々のもの(たとえば GoF のパターン)、Fowler が取り上げている企業内の多様な役割と社員の関係を扱う例題、などを試みている。

参考文献

- 1) M.Fowler. Dealing with roles. <http://www2.awl.com/cseng/titles/0-201-89542-0/apsupp/>. supplemental information to *Analysis Pattern*, Addison-Wesley, 1997.
- 2) W. Harrison and H. Ossher. Subject-oriented programming (a critique of pure objects). In *OOPSLA '93*, pages 411-428, 1993.
- 3) G. Kiczales, J. Lamping, A. Mendhekar, C. Maeda, C. Lopes, J.-M. Loingtier, and J. Irwin. Aspect-oriented programming. In *Proceedings of the European Conference on Object-Oriented Programming(ECOOP)*, Finland. Springer-Verlag, June 1997.
- 4) T. Tamai. Objects and roles: modeling based on the dualistic view. *Information and Software Technology*, 41(14):1005-1010, 1999.
- 5) N. Ubayashi and T. Tamai. An evolutionary cooperative computation based on adaptation to environment. In *Proceedings Asia Pacific Software Engineering Conference '99*, pages 334-341. IEEE, 1999.