

4U-8

設計モデル理解のための OMT シミュレータの検討

畑口剛之, 池田健次郎, 岸知二

NEC マイコンソフト開発環境研究所

1 はじめに

種々の状況における設計上の一般的な問題の解決に適用できるように、オブジェクトやクラス間の通信を記述したものをデザインパターンと呼んでいる [1]。このデザインパターンを利用して設計モデルを構築する方法は、設計の効率と安定性において有利であると我々は考えている。

しかしながら設計モデルがデザインパターンどおりに構築されているかどうかを確認する作業は必ずしも容易ではない。特に動的なふるまいがデザインパターンの主旨どおりに設計されているかの確認は困難なことが多い。

本稿では設計モデルを作成した段階で、設計モデルがデザインパターンどおりに構築されているかどうかの確認作業を設計シミュレータ [2] を用いて支援する手法を提案するとともに、実際の試行結果について報告する。

2 設計モデル理解に対する問題

本シミュレータは、OMT 法で記述された設計モデル（オブジェクト図と状態図）を入力としてコード生成してシミュレーションを行ない、オブジェクト間のメッセージのやり取り、オブジェクトの状態の変化などのログを出力するものである。これらのログには、シミュレータの実装に依存した時間軸における時刻が現れる。

デザインパターンでは、典型的な状況に対する動作をインタラクション図に記述することによってモデルの動的ふるまいを例示している。それに対

“Investigation of OMT Simulator for Understanding Design Model”, Takeshi HATAGUCHI, Kenjiroh IKEDA and Tomoji KISHI, NEC Corporation

して設計モデルでは、システムの動的ふるまいを状態図として一般化して定義している。このようにデザインパターンと設計モデルでは、示される動的ふるまいの抽象度や記法が異なる。

また、デザインパターンでは動作の論理的な半順序関係しか規定されていないのに対し、実行ログにはシミュレータの実装に依存した物理的な全順序関係が現れる。

従って設計モデルがデザインパターンの主旨どおりに構築されていることを確認するには、以下を確認することと考えられる。

- P1. デザインパターンで例示されている状況におけるふるまいが正しいかどうか。
- P2. モデルの論理的な半順序関係が正しいかどうか。

3 動的ふるまいの観測の支援方法

3.1 方針

以下、P1 と P2 の確認のための支援方法について考察する。

P1 については、デザインパターンで例示されている状況を作り出しやすくするための支援を、P2 については、物理的な全順序関係を含んだ実行ログから論理的な半順序関係を抽出するという方針をとった。

3.2 例示された状況の確認

デザインパターンで例示されている状況を作り出せば、その動作をシミュレータで確認できる。しかし、そのためにはシミュレータで例示されている状況を作り出すためのメッセージ群を指定しなければならず、その作業は必ずしも容易ではない。

“モデルの動的ふるまいのデザインパターンによる主旨”を確認するためには、デザインパターンで例示されている状況をより容易にシミュレートできることが望まれるので、シミュレーションに必要なオブジェクトに対して状態やオブジェクト間の関係について指定することにより、そのような状況が作り出せるような支援を行った。

3.3 半順序関係の確認

一般にデザインパターンでは、メッセージの遅延を0とみなした例示がされている。その結果、インタラクション図のメッセージの矢印は水平線となる。

一方、インタラクション図に全順序関係が含まれると、動作の論理的な半順序関係の確認の邪魔になる。図1, 2は、動作の論理的な半順序関係は等しいが、全順序関係が異なる実行ログをインタラクション図にしたものである。

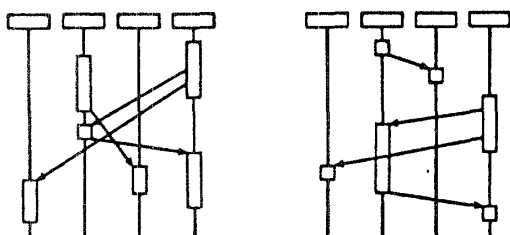


図1: 全順序を含む例1 図2: 全順序を含む例2

すべてのメッセージのやりとりを水平線のみで表示することができるわけではないが、デザインパターンでは水平線で表示できる例を扱っていることが多い。ならば全順序を無視し、半順序を保ちながら水平線表示すればデザインパターンとの対比が容易になると考えた。

4 試行結果

紙面の都合上、ここではP2の確認に対する支援の実行結果だけを紹介する。

あるデザインパターンに基づいた設計モデルをシミュレーションした実行ログを、忠実にインタラクション図に書き下すと図3のようになり、上記方

針で書き直すと図4のようになる。図4ではすべてのメッセージが水平線になっており、元のデザインパターンのインタラクション図によく似たものになっている。

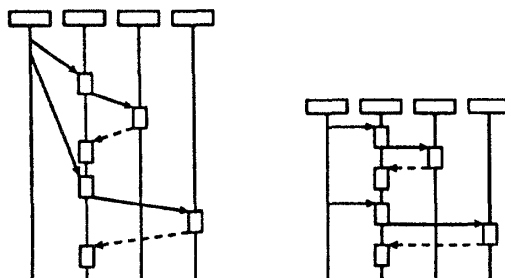


図3: ログに忠実な描画 図4: 本方法による描画

5 考察

3章で述べたように全ての実行結果を水平線だけで表現できるわけではないが、その場合でも斜め線が減少することによって対比が容易になると考えられる。

オブジェクトが増えたり、複雑なメッセージのやりとりをするようなモデルに対して、必ずしもこのような良好な結果が得られるとは限らないが、そのような場合には特定のオブジェクトが関わる処理に注目するための支援が有効であると考えている。

6 おわりに

以上、デザインパターンの動的ふるまいの観測を例にとり、OMTシミュレータを用いた設計モデル理解のための支援方法の考察を行なった。

今後は複数のデザインパターンが組み合わさって構成されているモデルに対して、設計モデルの理解を支援する方法などを検討する予定である。

参考文献

- [1] Erich Gamma, 他: デザインパターン. ソフトバンク, (1995).
- [2] 池田健次郎, 他: スパイラルなモデリングを実現する設計支援環境の提案, ソフトウェア工学研究会 オブジェクト指向 '96 (1996).