UML 設計に対するモデル検査のための検証パターン

金井　勇人†　　岸　知二†
北陸先端科学技術大学院大学†

1. はじめに

近年，組み込みソフトウェアは大規模，かつ複雑になってきており，従来の検証手法の限界が指摘されている．よって，経験則による手法だけでなく，形式的手法の導入が検討されている．形式的検証手法の１つにモデル検査技術[2]がある．この検証技術は，ソフトウェアの有限状態モデルが，論理式で表現された性質を満たすかどうかを状態の網羅的な探索によって検証を行う技術のことである．モデル検査技術をUML等で記述される設計モデルに適用することにより，ソフトウェアの信頼性を高めることが期待できる．しかし，UML 等で記述されたソフトウェアのモデルをモデル検査ツールで検証するためには，そのツールに依存した言語(以下，仕様記述言語と呼ぶ) で記述しなければならない．また，モデル検査技術ではソフトウェアが満たしてほしい性質を確認するために，性質を時間的な概念を持たせた論理式(以下，時相論理式と呼ぶ) で記述することも必要になり，ソフトウェア技術者にとって扱いづらい作業になる．

我々は，ソフトウェアのそれぞれの構造によって確認したい典型的な性質があり，またその確認方法にはいくつかの定石があることに注目し，それをパターンとして提示することで設計検証を支援を行う研究を進めている．今回は，モデル検査ツール SPIN[1] を対象とする．本稿は，以前我々が提案した構造付き検証パターン [3][4] の改良と評価について報告する．改良点は，パターン構造の改良である．

2. 構造付き検証パターンの提案

2.1 従来の代表的な検証パターン

形式的検証を支援するひとつの方法として，パターンを提示することが考えられる．現在までに提唱されている代表的な検証パターンとして，Dwyer らの検証パターン [6] がある．この検証パターンはよく使われる時相論理式の記述を性質ごとに分類しパターン化している．

Verification Patterns for Model Checking Software Design
†Japan Advanced Institute of Science and Technology

2.2 Dwyer らの検証パターンの特徴と問題点

Dwyer らの検証パターンは対象システムを特定していないので，汎用的に利用できるという特徴を持つ．問題点として，汎用的なので特定のソフトウェア構造上の具体的な性質の表現にどう利用するかが明確に示されていない点が挙げられる．

2.3 本提案のねらい

本稿で提案する検証パターンの特徴は UML の特定の構造に対して，その構造において重要な性質をリストアップし，それぞれの性質を検証するための LTL 式のパターン化を行っている点である．これによりソフトウェア技術者にとって利用しやすいパターンとなることが期待される．本検証パターンの構成は以下のものである．

- UML から PROMELA への変換規則
- 検証パターンカタログ
  - 設計モデルに多出する構造を抽象化した構造
  - 上記の構造でよく検査される性質を LTL 式で記述したもの

本検証パターンを利用した想定される典型的な手続きを図１に示す．

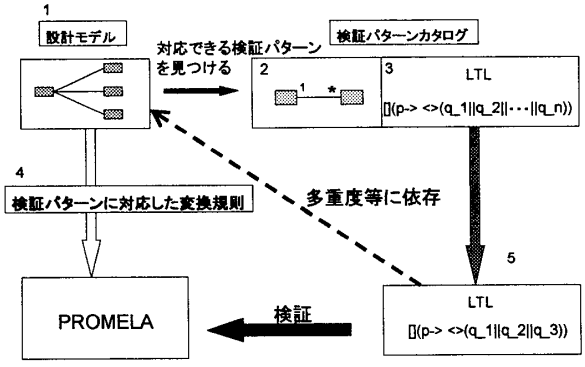

図１　典型的な検証手続き

2.4 構造付き検証パターンの一覧

ソフトウェア構造の基本性質のパターンは企業から提供された事例を分析して，そこで多出する構造と必要な検証項目に注目してパターン化したものである．以下の構造に対して構造付き検証パターンをまとめた．

（１）2 オブジェクト間のメッセージ送受信
（２）オブジェクト間連続メッセージ送受信
（３）2 オブジェクト間の属性値

3. 評価

構造付き検証パターンの妥当性について評価を行った．Real-Time Design Patterns(RTDP)[5]は，組込みソフトウェアでよく使われる構造をパターン化したものである．各パターン毎に例題が載っており，それは組込みソフトウェアで実際によく利用されるソフトウェア構造と言える．この RTDP の例題に対して，本検証パターンの静的構造と性質の適合性を調べた．RTDP の例題にはステートマシン図による動的構造が記載されていないので，今回は動的構造の適合性調査は行わない．

➢ 静的構造との適合

表１　2 オブジェクト間のメッセージ送受信

パターングループ	対象数	適用数
Concurrency	7	3
Memory	6	0
Resource	6	0
Distribution	6	6

表２　オブジェクト連続メッセージ送受信

パターングループ	対象数	適用数
Concurrency	7	1
Memory	6	0
Resource	6	0
Distribution	6	1

表３　2 オブジェクト間の属性値

パターングループ	対象数	適用数
Concurrency	7	1
Memory	6	0
Resource	6	6
Distribution	6	5

➢ 検証性質との適合

表４　2 オブジェクト間のメッセージ送受信

パターングループ	対象数	適用数
Concurrency	3	3
Distribution	6	6

表５　オブジェクト連続メッセージ送受信

パターングループ	対象数	適用数
Concurrency	1	1
Distribution	1	1

表６　2 オブジェクト間の属性値

パターングループ	対象数	適用数
Concurrency	1	1
Resource	6	6
Distribution	5	5

4. 考察

RTDP は，３つの粒度に分類し，パターン化している．そのうち Memmory Patterns は細粒度の詳細設計の分類であるが，提案する検証パターンはこの粒度の例題には適用できなかった．一方，中程度の粒度であるメカニズム設計に該当する Concurrency Patterns, Resource Patterns や，大粒度であるアーキテクチャ設計に該当する Distribution Patterns に対しては多くの例題が適用できることがわかった．上記より，本検証パターンは中粒度，大粒度のソフトウェア構造に対して有効性があると考えられる．なお，2 オブジェクト間のメッセージ送受信パターン，2 オブジェクト間の属性値パターンでは，多くの例題に対応できた．しかし，オブジェクト間連続メッセージ送受信パターンは，多くの例題に対応することができなかった．これは，対象にした例題の中に，3 つ以上の関連したメッセージ送受信構造が少なかったためだと考えられる．これについては今後もっと大きな例題に対して評価をしていきたい．

本検証パターンのパターン構造はクラス図とステートマシン図から構成されるが，その表記は特定の具体的なひとつの構造を表すのではなく，その検証手法が適用できる構造の持つ制約を示すものでなければならない．今回の構造表記法では，この制約があいまいになっている．明確な制約を記述するような構造の表記法を今後提案していきたい．

参考文献

1) G. J, Holzmann. : The SPIN Model Checker. Addison-Wsley 2004.

2) E. Clarke, O. Grumberg, and D. Peled : Model Checking, MIT 1999.

3) 金井勇人，岸知二：UML 設計モデル検査技術のための検証パターンの提案，情報処理学会ソフトウェア工学研究会，SE152-3, pp17-24, 2006.

4) 金井勇人，岸知二: UML 設計に対するモデル検査のための検証パターンの提案と評価，情報処理学会ソフトウェアエンジニアリングシンポジウム，pp185-192, 2006.

5) Douglass, Bruce Powel. : Real-Time Design Patterns. Addison-Wsley 2003.

6) Matthew B. Dwyer, George S. Avrunin and James C.: Patterns in Property Specificationsfor Finite-state Verification, the 21st InternationalConference on Software Engineering, May, 1999