

検証ルールによる拡張可能な UML 図間の整合性検証ツール

名川裕樹[†] 小形真平[‡] 香山瑞恵[†] 海尻賢二[†]

信州大学工学部情報工学科[†] 信州大学大学院理工学系研究科[‡]

1 はじめに

ステレオタイプにより拡張された UML モデルを用いた、様々な設計手法が提案されている。しかし、モデルの整合性を手動で検証する場合は、時間がかかる。また、設計手法ごとに独自の拡張がなされ、整合を取る箇所は異なるため、手法ごとに整合性検証ツールを開発することは、開発者の手間となる。本研究では、ステレオタイプによる拡張に柔軟に対応できる UML モデルの整合性検証ツールを実現する。

2 整合性検証ツールの提案

本ツールでは検証を図 1 の手順で実施する。

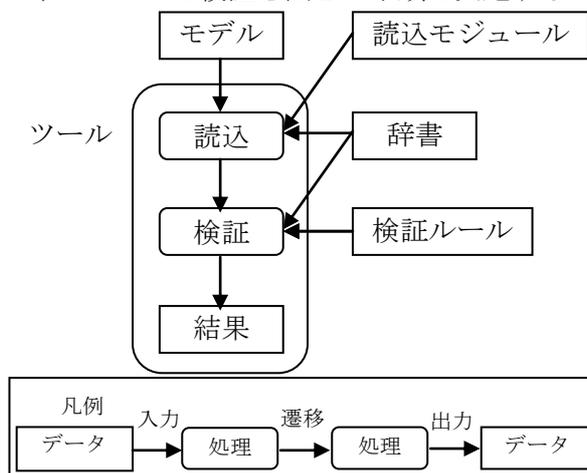


図 1 提案する整合性検証ツールの処理フロー

UML 図は拡張されてもグラフ構造になるため、名前などの情報を持つノード・エッジなどのグラフ要素を有するグラフ間の整合性を検証できれば良い。そこで、本ツールの処理を以下のように構成する。

読み込み: 利用者が検証対象の図要素名・ステレオタイプを記述した辞書とモデルをツールに入力する。ツールはその辞書を介して、モデルとツールのグラフ要素を対応付けて読み込む。ここで図要素とは、例えばクラスや状態などである。

検証: 辞書を語彙に後述の検証ルールに基づいて、ツールがモデルに対応するグラフ間の整合性検証を実施する。

A tool for verifying consistency between UML diagrams by using extendable rules.

[†]Yuki Nagawa, [‡]Shinpei Ogata,

[†]Mizue Kayama, [†]Kenji Kaijiri.

[†]Department of Computer Science & Engineering, Faculty of Engineering, Shinshu University.

[‡]Graduate school of Science and Technology, Shinshu University.

図 2 は辞書の例であり、クラスやステレオタイプの boundary が記述された図要素などをノードとして扱い、関連やメッセージなどをエッジとして扱うよう記述されている。ステレオタイプは先頭に“S:”を付けて判別する。

また、読み込みモジュールとは、UML エディタごとに異なる図要素の取得方法に対応するためのものである。Java で実装した本ツールに“図要素取得用のインターフェイス”を実装したクラスを読み込ませることで実現する。

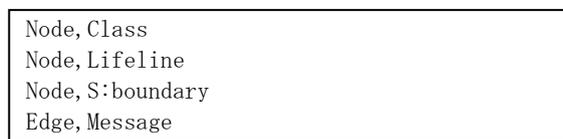


図 2 辞書の記述例

3 検証ルール

本検証ルールは、天野ら[1]の提案した存在チェックと一致チェックの2種類を検証内容として記述する。存在チェックは、2つの図で同名の要素が存在するかを検証する。一致チェックは、名前前で対応付けられた図要素の組に対して、さらに各図要素が持つ下位構造の指定された要素が同名かを検証する。例えばクラスとライフラインを組として、クラスが持つ操作名とメッセージ名が一致するかを検証することができる。

検証ルールは図 3 の“,”区切りの文字列である。

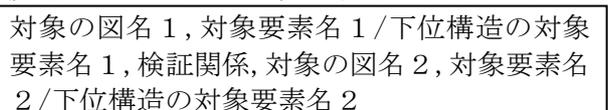


図 3 検証ルールの構文

表 2 に検証ルールの例を示す。下位構造の対象要素名が記述される場合は一致チェックとなり、記述されない場合は存在チェックとなる。

対象要素名は、2章で辞書に記述した要素名と、図 4 のクラス構造に基づいた属性名と関連端名を利用し記述する。記述方法は、辞書に記述した要素名を先頭に、対応するグラフ要素から参照可能な属性か関連端名を“.”で繋ぐ。表 2 (c) を例にすると対象要素は Message が辞書により Edge クラスと対応し，“.”で繋がれた関連端名 to の Node 名となる。

検証関係は、2種類から選択できる。互いの図に同名の対象要素が存在すべきならば“<->”, 対象要素 2 の集合が対象要素 1 の集合に含まれ

るべきならば“->”と記述する。

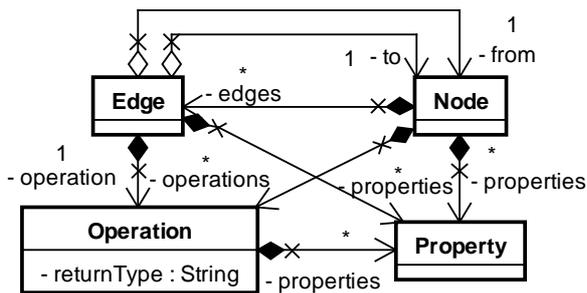


図 4 対象要素として記述できる要素名

4 評価

ステレオタイプによる拡張に柔軟に対応し、汎用的な整合性検証を本ツールで実施可能なことを示すため、次の2点を評価する。

- ・設計手法に応じた整合性検証を実施できるか
- ・既存研究の整合性検証内容を再現できるか

提案手法を適用する対象は、研究室で開発された、研究室内で共有される菓子類の在庫情報を管理する Web アプリケーションのモデルである。

モデルは画面遷移モデル[2]と、アプリケーションの簡単な処理フローを表現したクラス図(CD)とシーケンス図(SD)で構成される。画面遷移モデルは CD とステートマシン図(SMD)で構成される。画面遷移モデルの CD は、boundary のステレオタイプが記述されたクラスで、その画面の入出力項目や遷移項目が記述される。SMD は画面間の遷移が記述される。

図の数は画面遷移モデルで CD 2 個、SMD 2 個、CD と SD の整合性検証に利用する CD は 7 個、SD は 9 個である。検証の項目数と検証規則の数は表 1 の通りである。

表 1 検証項目と検証規則の数

モデル	検証の項目数	検証規則の数
画面遷移モデル	2 項目	4 個
CD と SD	3 項目	27 個

画面遷移モデル独自の検証項目として、(a)CD のステレオタイプ boundary のクラスについて、クラスと同名の状態が SMD に存在するか、(b)SMD 中の遷移のトリガーと同名のクラスの属性が存在するかを検証する。CD と SD では既存の整合性検証手法[3]から、(c)SD 中のメッセージと同名のクラスの操作が CD 中に存在するか、(d)SD でライフライン間のメッセージがあるとき、CD で対応するクラス間に関連があるか、(e)SD 中のメッセージの引数が、CD 中の操作の引数と同じかを検証する。表 2 は検証項目を検証規則として記述した一部である。提案の試作ツールでは、表 2 (b)の検証において edges をトリガーとして扱うようにしている。トリガーなどの

property を個別に指定できるように辞書などを拡張することは今後の課題となる。

検証規則は、検証する図の組ごとに記述する必要があるため、同項目の検証でも検証規則の数は複数になる。

表 2 検証規則として記述された検証項目例

項目	検証規則
(a)	CD 名, boundary, <->, SMD 名, State
(b)	SMD 名, State/edges, ->, CD 名, boundary/properties
(c)	CD 名, Class/operations, ->, SD 名, Message.to/operation

検証した結果、画面遷移モデルでは 2 個の検証規則で 4 箇所、CD と SD では 15 個の検証規則で 47 箇所の不整合を検出した。図 5 は結果の一部である。そして、各検証項目で見えされるべき不整合を漏れなく正しく検出したことを確認した。

```
CD1, boundary, <->, SMD1, State: error
パスワード変更確認(管理者)が図名:CD1に存在していません
パスワード変更確認(購入者)が図名:SMD1に存在していません
```

図 5 整合性検証結果の一部

5 おわりに

本論文ではステレオタイプにより拡張されたモデルに対して整合性検証を実施するツールと検証規則を提案した。本提案により検証規則を書くだけで、一つのツールで様々な整合性検証が可能になる。

今後の課題として、4章にあげた property の個別指定の問題改善のために、辞書と検証規則を拡張し、辞書で edge の property と trigger を対応させ、検証規則の properties と記述可能な箇所に trigger と記述すると、検証時に properties 要素中の trigger 要素を対象要素にすることを可能にする。また、検証規則や辞書などのカスタマイズの手間と手法ごとに検証ツールを開発する手間の比較評価を実施する。

6 謝辞

この研究の一部は JSPS 科研費 22300286 の助成を受けたものである。

参考文献

- [1]天野 富夫, 石川 雄一, モデルとパターンに基づく静的テスト仕様記述の自動生成, IBMPROVISION No. 67, pp94-100, 2010.
- [2]紙森 翔平, 小形 真平, 海尻 賢二, モデル駆動開発におけるユーザビリティ機能を実装した Web プロトタイプの自動生成, IPSJ/SIGSE SES2014, pp. 146-151, 2014.
- [3]大西 淳, UML におけるモデル整合性検証支援システム, 電子情報通信学会論文誌 D-I Vol. J84-D-I No. 6, pp. 671-681, 2001.