B-001

自動コード生成を目的とした 複数の UML 図に対する整合性検査手法

Integrity Checking Method for Multiple UML Diagrams for Automated Code Generation

畑瀬 尚之 †　　和崎 克己 ††
Takayuki Hatase　Katsumi Wasaki

1　はじめに

ソフトウェア開発の早い段階でのプロトタイピングは，要求された仕様の確認を可能とするものであり，欠陥発覚の遅れに対して有用な手段である．このとき，形式手法を用いた上位設計群を活用することで，より信頼性のあるプロトタイピングとなると考える．しかし，複数の形式手法による上位設計間で整合性が取れていなければ，そこから得るプロトタイピングも誤ったものとなってしまう．本研究ではプロトタイピングを行う上で用いられる上位設計群間での整合性検査を行い，より信頼性の高い自動コード生成を目指している．

2　上位設計群を用いた自動コード生成系

2.1　準備研究の状況

本研究の最終的な目的は上位設計群を利用しての自動コード生成であるが，その準備研究として VDMJWeb サービス [1] と Java Servlet スケルトンコード生成器 [2] が挙げられる．VDMJWeb サービスは，インタプリタ機能を有する VDMJ モデル実行器と，VDMJ のクライアントとして外部と通信を行う機能が備わった VDMJC から構成される．ユーザは Java Servlet により Web ブラウザに表示された処理結果を見て妥当性確認を行うができる．また，Java Servlet スケルトンコード生成器は，イベントに対しユーザが予約を行うという単純な予約システムを対象とした UML アクティビティ図を用いての画面遷移条件およびロジックフローを記述し，その成果物を用いて VDM++ で記述された仕様と一対一に対応したツールである．

2.2　複数の準形式手法による上位設計

2.1 節で示した準備研究の手法では上位設計群の間での整合性確認が行われていないため，用いる上位設計群に誤りが混入した場合，誤り混入，生成不可という状況が起こりえる．このため，用いる上位設計群に対して整合性検査を行わなければならない．

3　複数の UML 図に対する整合性検査

3.1　提案手法

上位設計群間での整合性検査として対象を，システムの各要素の関係，メソッド等を記述したクラス図と，クラス図で記述されたメソッド等の実行系列を記述したアクティビティ図を自動コード生成に必要な UML 図として整合性検査を行う．また，システムの仕様記述となる

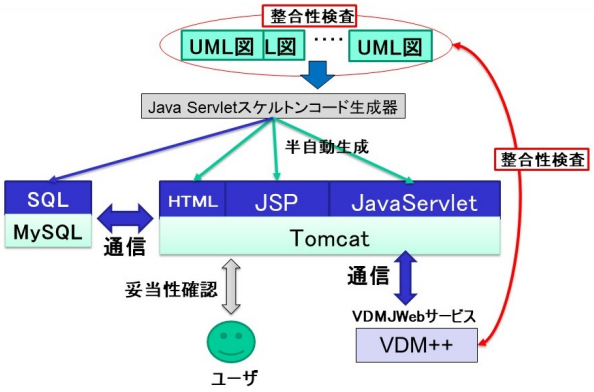

図 1　上位設計群に対する整合性検査

VDM++ ファイル群に対しても UML 図との整合性検査を行う．図 1 に従来研究である VDMJWeb サービスと Java Servlet スケルトンコード生成器による早期の妥当性確認手法に提案手法を加えたものを示す．

3.2　UML 図要素に対する一致検査

クラス図にはクラス名，インスタンス，関数・メソッドとその引数・返し値を記述し，アクティビティ図にはそれらの実行系列を記述する．このときクラス図とアクティビティ図間で記述された要素が一致するかの検査を行う．このとき，クラス図を設計としてアクティビティ図よりも上位のものであるとし，クラス図基軸による検査となる．この要素一致検査に際して，最低限満たすべき構造，検査を行う上での曖昧さの排除といった目的のため記述規則を設ける．要素の一致検査に加え，これらの記述規則が満たされているかも確認していく．記述規則は以下の通りである．

クラス図

(i)　最低 1 つのクラスを持つこと
(ii)　変数の記述は VDM++ でのインスタンス宣言と同様に記述する

アクティビティ図

(i)　最低 1 つのパーティション，開始・終了ノードを持つこと
(ii)　パーティションにクラス名，アクティビティノードにメソッド名，その入力フローのアクションに引数，出力フローのガードに返し値を記述すること
(iii)　引数をもたない場合，代わりに [no argument] と記述すること

† 信州大学総合理工学研究科，Graduate School of Science and Technology, Shinshu University.
†† 信州大学工学部，Faculty of Engineering, Shinshu University.

表 1　不整合内容に対応するユーザフィードバック例

評価値		不整合内容	ノート内容
1.Critical	11	図内にクラスを一つも持たない	Unable to find the Class in this ClassDiagram
	21	アクティビティ図の要素がクラス図に無い	Unable to find the [element] in ClassDiagrams
	31	ノードが途中で途切れている	Flow is broken at [node]
	41	分岐・合流/ループの開始と戻りの不一致	Unable to match any node with [node]
2.Warning	11	クラス図の要素がアクティビティ図に無い	Unable to find the [element] in ActivityDiagrams
	31	開始から終了の間で出現しないノード	[node] does not appear until FinalNode

3.3　UML アクティビティ図に対する構造検査

アクティビティ図には，クラス図で記述されたメソッド等の実行系列を記述する．このとき，生成言語によって記述可能となる構造も変わる．C++ 言語では goto 文による指定ラベルへの移動が可能だが，Java 言語では記述できない，といった具合である．このため，対象言語により検査項目も変わるが，現在の提案手法では，準備研究である VDMJ サービスや Java Servlet スケルトンコード生成器の利用を考えているため，対象言語を Java 言語とし，それに合わせた検査項目となる．この構造検査に際しての記述規則は以下の通りである．

(i) ループ条件による分岐のフローのガードに true/false を記述すること
(ii) ループの戻りは直接接続し，その間に他のノードを持たないこと

また，アクティビティ図の記述の際，使用するノードまたは表現する構造に対する入出力フロー数との対応は以下の通りである．

- **開始ノード** 入力 0，出力 1
- **終了ノード** 入力 1，出力 0
- **アクションノード** 入力 1，出力 1
- **フォークノード** 入力 1，出力 2 以上
- **ジョインノード** 入力 2 以上，出力 1
- **条件分岐** 入力 1，出力 2 以上に対し，入力 2 以上，出力 1
- **do-whileloop** 入力 1，出力 2 に対し，入力 1，出力 2
- **for/whileloop** 入力 2，出力 2 に対し，入力 1，出力 1

条件分岐やループ構造はディシジョン/マージノードを用いて記述される．このとき分岐ノードと合流ノードでの対と考える．例えば，条件分岐に対するフロー対応は，入力が 1，出力が 2 以上となる分岐ノードに対して，入力が 2 以上，出力が 1 となる合流ノードが対となるとして表現する．

4　上位設計ツールへの取り組み

4.1　検査結果のユーザへのフィードバック

検査結果をユーザにフィードバックし，記述のサポートを行う．フィードバックとして，プロジェクト全体での不整合要素の要素名，型等の出力と，表 1 で示すような不整合内容に対応したエラー文を不整合要素と紐づけた形で図上に生成する．これによりユーザは不整合要素に修正を加え，再度検査を行うことで徐々に整合性のとれたモデルに近づけていくことを想定している．

4.2　astah*プラグインとしての実装試行

提案手法による整合性検査とそのフィードバックを astah*professional[3] にプラグインとして実装を行った．astah 上でのフィードバックの例を図 2 に示す．

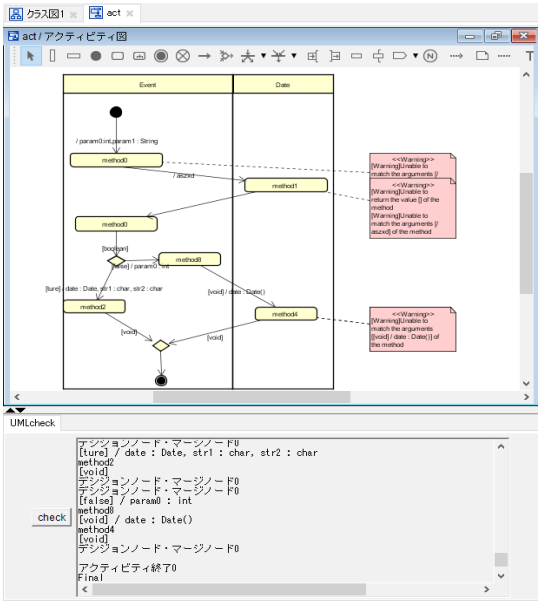

図 2　検査結果のユーザフィードバック例

図 2 のアクティビティ図は検証用にいくつかの不整合要素を挿入したものである．例えば，ノード method4 の引数である date： Date() はクラス図で定義された method4 の引数と一致していない．そのため，フィードバックとして対応したエラー文がノートとして生成される．このとき，示された不整合要素に対して正しく修正が行われれば，次の検査時にはノートが削除される．

5　まとめと今後の課題

自動コード生成を目的とした上で利用される上位設計群の整合性検査の提案と検査部の一部実装を行った．今後の課題として，現在の提案手法では記述規則が非常に厳しく，対応する構造が少ない．そのため，例外処理の追加やコネクタによる図間接続，言語別の検査項目等のサポートを行うべきである．また，記述規則の厳しさによって提案手法での UML 図の記述が難解なものとなることが予想されるため，生成対象を絞り対象別のデザインパターンの導入などが考えられる．

参考文献

[1] 村林慧，多田圭佑，和崎克己：“VDMJ と Apache Axis2 を用いた上流工程におけるモデル実行環境の構築，”FIT2014 (第 13 回情報科学技術フォーラム) 講演論文集，(B-108)，pp155-158，2014
[2] 森島耕，和崎克己：“テストコード半自動生成を用いたプロトタイプ検査と VDM++ 妥当性確認のコスト削減，”FIT2016 (第 15 回情報科学技術フォーラム) 講演論文集，(A-010)，pp.103-106，2016
[3] 株式会社チェンジビジョン：“astah*professional，”http://astah.change-vision.com/ja/product