VDM++仕様から C#コードを生成するツールの開発と評価

千坂 優佑　　大友 楓雅　　力武 克彰　　岡本 圭史
仙台高等専門学校
{a1502020, a1602006}@sendai-nct.jp　　{yoshiaki, okamoto}@sendai-nct.ac.jp

要旨

本報告では，VDM++仕様から C#コードを生成するツール（以下，本ツール）の開発と妥当性・有用性評価について報告する．本ツールを用いることで，VDM++仕様から C#コードを生成でき，CodeContracts による事前・事後・不変条件を用いた生成コードの検証が可能となる．

1. はじめに

情報システムの信頼性向上の有効な手段として，形式手法がある．形式手法の 1 つである形式仕様記述は，仕様を VDM++等の形式仕様記述言語で記述することで，仕様の厳密化や計算機による検証を可能とする．

VDM++を扱う既存のツールには， VDM++で記述された仕様から Java または C++のコードを生成する機能がある．この機能により，仕様を基にコードを作成する工数を削減するだけでなく，その過程でのヒューマンエラーの混入を予防できる．しかし，現状のコード生成機能では Java と C++しか出力できず，また操作定義の事後条件は変換できないなどの制約もある．

本報告では，VDM++仕様から C#コードを生成するツールの開発について報告する．本ツールが VDM++仕様における事前・事後・不変条件を生成コード中の契約へ変換し，C#から利用可能な CodeContracts を用いることで，検証に事前・事後・不変条件を利用できる．

2. コード生成の方法

コード生成処理は，構文解析，抽象構文木の変換，C#コード生成の 3 つの手順で行われる．構文解析および C#コード生成の実装には，既存のツール VDMJ および.NET コンパイラープラットフォーム Roslyn を利用する．

VDM++仕様は 1 つ以上のクラスから成り，クラスはいくつかの定義ブロックから成る．それらの要素ごとに対応する C#の要素を定め，変換を行う．VDM++の事前・事後・不変条件は，CodeContracts においてそれらを表す Contract.Requires・Contract.Ensures・Contract.Invariant に変換する．VDM++の事後条件において返り値を表す RESULT は，メソッドの返り値を表す Contract.Result に変換する．また，操作定義の事後条件において変数の旧値にアクセスするために用いられる旧名称は，Contract.OldValue に変換する．

表 1. コード生成結果の例（一部）

post Count = Count~ + 1;
Contract.Ensures(Count==Contract.OldValue(Count)+1);

3. 本ツールの評価

本ツールによるコード生成の妥当性確認のため，VDM++仕様と本ツールにより生成された C#コードの等価性を調査した．具体的には，簡単な VDM++仕様に対して仕様アニメーションによるテストを行い，本ツールが生成した C#コードに対しても同じ事前・事後条件を用いて単体テストを行ったところ，それぞれ同一の入力に対して同一の出力が得られ，同一のテスト結果が得られた．

次に本ツールの有用性を評価するため，3 つのクラスから成るシステムを対象として従来開発(A)および本ツールを用いた開発(B)を実施し比較した．(A)では，作成済みの UML モデルに基づき 18 行の C#スケルトンコードを作成するのに 15 分程度の時間を費やしたが，(B)では 1 分もかからずに生成できた．さらに，(A)では UML に事前条件が記述されていなかったため事前条件を満たさないような入力も含めたテストケースが作成されたが，(B)では CodeContracts により事前条件が保持されていたためそのようなテストケースは作成されなかった．

4. まとめ

本報告では，VDM++仕様から C#コードを生成するツールを開発し，その妥当性・有用性評価を行った．本ツールを用いることで，開発期間が削減された．また，優先度の高いテストケースに集中できることが簡単な例題で確認できた．