B-008

UML/OCL から SQL への変換

Conversion from UML/OCL to SQL

黒澤 慎太郎† 小林 洋†
Shintarou Kurosawa Hiromi Kobayashi

１．はじめに

最近のソフトウェアの設計では、表明(assertion)の記述が重視されるようになって来ている。オブジェクト指向のソフトウェアのシステム設計では、図式仕様の UML が広く用いられているが、表明を正確に記述するには論理を用いる必要があり、UML では OCL(Object Constraint Language)などで図式仕様に補足的に記述することになる。一方、データベースアプリケーションの設計では ER 図が普及しており、ER 図を基に SQL で実装を行うということが良く行われている。この実装において SQL92 規格では表明が制定されているが、製品レベルでは表明の実装は進んでおらず、トリガ(Trigger)や制約(check)を用いて実装しているのが現状である。そこで本研究では、データベースを用いるソフトウェア開発において、UML のクラス図と OCL で補足的に記述した表明を、トリガに変換するための方式についての研究を行った。なお、データベースとしては PostgreSQL を使用した。

２．変換方式の概要

OCL は制約条件を正確に表すために制定された言語である。OCL から Java 言語で実装する際の変換については既にいくつかの研究が行われているが[1-3]、業務系アプリケーションの開発において通常用いられているデータベースの SQL への変換についてはあまり研究が見当たらない。SQL92 規格では CREATE ASSERITON 文により表明が定義できるように制定されているが、DBMS の製品ではこの実装は進んでおらず、トリガを用いているのが実情であるため、本研究ではトリガに変換することにした。トリガの記述は製品によって多少異なるところがあるが、本研究では PostgreSQL を使用した。なお、本方式では図１に示すように、OCL については SQL のトリガに変換すると共に、クラス図については既知の方法で ER 図の一種であるバックマン（Backman）図に一旦変換した後、SQL のテーブルに変換することにする。

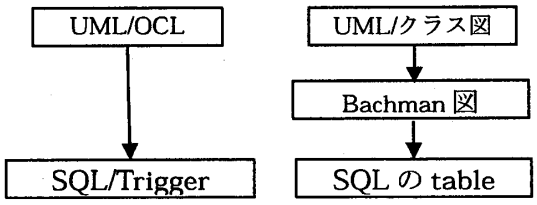

図１　UML から SQL への変換

†東海大学 工学研究科, Tokai University

３．変換手順

３．１　ＳＱＬのテーブル作成

クラス図をバックマン図に変換し、これを基に、SQL のテーブルを作成する。この際、テーブル名は、クラス図のクラス名と同じとする。

３．２　ＯＣＬ記述

OCL で表明を記述する際には、テキスト形式でコメントとしてクラス図に図２に示すように記述することにする。

(1) Context には、クラス、属性、操作のいずれかの名前を記述する。属性名又は操作名を記述する場合には、それらが属するクラス名も併せて記述する。

(2) 記述する表明としては、事前条件(pre)、事後条件(post)、不変条件(inv)のいずれかとする。一つの OCL に複数の表明を記述することも出来る。

(3) 制約式としては、四則演算子と比較演算子にこれらを結ぶための and、or を用いる。

```
Context [クラス名 | 操作名 | 属性名]
[pre | post | inv] : 制約式...
```

図２　OCL 記述の枠組み

３．３　トリガへの変換

OCL から SQL のトリガへの変換を行う。図３にトリガ文の枠組みを示す。なお、トリガ名と関数名には任意の名前を記述する。

（１）　実行条件の変換

①　[BEFORE | AFTER] : OCL で記述されたアサーションの事前条件(pre)と事後条件(inv)に対して、それぞれ事前条件には BEFORE を、事後条件には AFTER を割り当てる。BEFORE ではデータベースで行われる操作の処理前に、AFTER では処理後にトリガを実行する。不変条件(inv)については、操作に対して BEFORE と AFTER を両方行うことで対応する。

②　命令後指定：トリガで、データベースの操作である挿入(INSERT)、更新(UPDATE)、削除(DELETE)が行われた際の、処理の対象となる操作を記述する。なお、or を用いて、対象となる操作を複数記述することも可能である。

```
1:    CREATE TRIGGER ‘ トリガ名’
2:    {BEFORE | AFTER}
3:    [命令語指定]
4:    ON [テーブル名]
5:    EXECUTE PROCEDURE [関数名] ()
6:    CREATE FUNCTION [関数名 ]();
7:    RETURNS TRIGGER
8:    LANGUAGE PLPGSQL AS’
9:    DECLARE
10: [変数宣言]
11: BEGIN
12: [SELECT 文]
13:      [式]
14:   END ’ ;
```

図３　Trigger の枠組み

③　テーブル名：トリガで操作の対象とする、データベースのテーブル名を記述する。OCL 記述の Context で指定されているものを次のように変換する。Context でクラス名が記述されていれば、そのままテーブル名として用いる。Context で属性名又は操作名が記述されているときには、属性または操作が属するクラスで用いるようにする。

（２）　式の変換

OCL で記述される条件で、データベースが持つテーブルの値を参照する場合には、図４のような SQL 機能の SELECT 文を用いることにより実現する。SELECT 文を使って値を参照するときには、その値を入れる変数を宣言する必要がある。変数名は任意とするが、変数型は参照する値と同じ型を用いる。但し、図４は、検索結果が１行に限定される場合で、検索結果が複数行からなる一般的な場合には、この枠組みではなくカーソル(cursor)を用いた枠組みを用いる必要がある。

```
1:  SELECT [属性名] FROM [テーブル名]
2:  INTO [変数名]
3:  WHERE [検索条件]
```

図４　SELECT 文の枠組み

本方式で変換する OCL の制約条件式には、(a) 値に演算処理を行うものと、(b) 等号や不等号を用いて値を比べ、式の条件を満たしているか判断するものの２つのパターンがある。前者では、トリガの式で演算処理を行えるように記述する。後者の場合には、図５のようにトリガに IF 文を用いることで実現する。但し、図５の条件式は、OCL で記述された制約式を否定したものである。つまり、OCL の制約条件式を満たしていない場合には、図５の IF 文の

```
1:  IF
2:      [条件式]
3:  THEN
4:      NEW := NULL;
5:   END IF
```

図５　IF 文を用いた式

条件式が成り立つことになり、その場合には、操作で変更されたデータを４行目の[NEW := NULL;]により無効としている。反対に OCL の制約式を満たしている場合には、IF 文の条件式が成立しないことになり、データベースへの操作はそのまま行われることになる。

４．制約式から条件式への変換

OCL で記述された制約式からトリガの条件式に変換する処理においては、条件式を以下のように token に分解することにより変換を行っている。

①　OCL の制約式に記述されている数字、データベースから参照された値、四則演算記号、比較演算記号、及び and、or を、それぞれ式の左から順に token[0]...token[n] とする。

②　変換先のトリガの条件式には、左から token[n]...token[0]と反対に並べ記述する。

③　IF 文での条件式は、OCL の制約式を否定したものであるため、比較演算子を含む token では、それぞれ反対の意味を持つ比較演算子に置き換える。

５．おわりに

本稿では、UML/クラス図に OCL で記述した表明を現状の実装されている SQL で実現するために、トリガに変換する方式を提案した。今回開発している方式では、まだ制約式は四則演算子と比較演算子及びその and、or に限られている。また、本方式では、OCL の制約式を満たさない場合、トリガの IF 文で操作を無効にするのみであるが、この場合の処理方法については、今後の課題と考えている。現状では、本稿で示した方法の一連の変換は手作業で行っているが、将来的には、各記述の枠組みを用いて、OCL からトリガへ変換する開発支援ツールの作成を考えている。

参考文献

[1]　ヨシュ・ヴァルメル，アーネク・クレッペ: UML/MDA のためのオブジェクト制約言語 OCL，株式会社エスアイビー・アクセス (2004).

[2]　牧寺彩，長井栄吾，岡野浩三，谷口健一:UML/OCL を用いた分散実時間アプリケーション開発手法の提案,信学技報，Vol.104，No.723，SS2004-64，pp.1-6 (2004).

[3]　A. Hamie : Translating the Object Constraint Language into the Java Modelling Language，SAC04，(2004).