

製品モデル表現におけるEXPRESSとJavaの言語機能比較

北海道大学大学院工学研究科 ○佐藤 友紀 田中 文基 岸浪 建史

要 旨

ISO10303 Part11 に定められている形式言語 EXPRESS は、人間の可読性を高めるための図式表記法 EXPRESS-G をサブセットとして持つオブジェクト指向型言語であるが、実装機能を持たないため他の言語に置換えて実装しなければならない。一方実装言語 Java はサブセットとして図式表記法 Java-g が提案されており、EXPRESS 同様オブジェクト指向型言語であるので、モデルの形式表現とその実装を同時に行える可能性がある。従って、本研究ではこれらの言語の形式言語としての言語機能比較を行う。

1.はじめに

EXPRESS[1]は形式言語であり、対象を表現しその意味を検討する上でも広く利用されている図式表記法EXPRESS-G[1]を持っている。しかし、実装機能がないため図1にあるようにEXPRESSで記述したモデルを実装するにはSTEPツールを用いてCやC++言語に置換えなければならない。その際に情報の欠落が生じてしまうという問題点がある。一方、実装言語Java[2]は、オブジェクト指向型言語であること、及び図式表記法Java-g[3]を持つことから、EXPRESSと同機能の形式言語となる可能性を持っている。もし、モデルの形式表現が可能であれば、本来実装言語であるJavaは図1にもあるように変換することなく、その実装をも同時に行えるという利点がある。よって本研究は以下の目的の下でEXPRESSとJavaの形式言語としての言語機能比較を行うものである。

1. 形式言語としての観点から EXPRESSとJavaの機能及び用語の対応関係を明らかにする
2. それらのサブセットである図式表記法 EXPRESS-GとJava-gの対応関係を明らかにする

言語の種類	形式言語		仕様から実装への変換	実装言語
	図式表現	言語表現		
EXPRESS	EXPRESS-G	EXPRESS	入力 STEP-Tool 変換	C++ / C
Java	Java-g	Java	無変換	Java

図1 EXPRESSとJavaの言語機能範囲

2. EXPRESSとJavaの言語構造

EXPRESSの言語構造を図2に示す。EXPRESSでは物をエンティティと呼ばれる型で表現する。エンティティは多重継承の種類を定める下位型拘束によって上位型から下位型へと性質を継承させることが可能である。その性質はデータ型によって定義される属性と、それに対する制約によって定義する。次に、Javaの言語構造を図3に示す。Javaでは物をクラスという型で表現し、単一継承によってのみ上位型から下位型へと性質を継承させることが可能である。その性質はデータ型によって定義されるフィールドと、そのフィールド及びクラス自体に制約を与える修飾子と、フィールドを操作

するメソッドによって定義する。

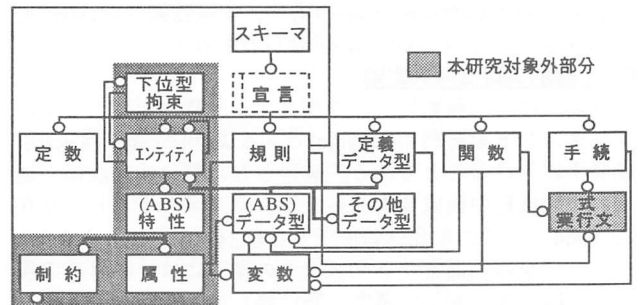


図2 EXPRESS-GによるEXPRESSの言語構造

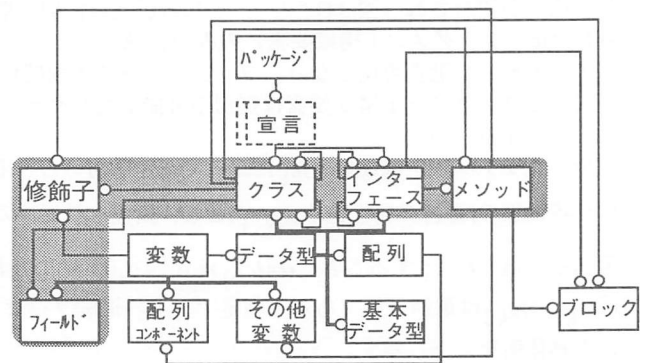


図3 EXPRESS-GによるJavaの言語構造

3. 言語機能比較

2.で述べたように EXPRESSとJavaでは、物を表現するための類似した構文要素(予約語、Reserved words)をもった言語であることは明らかである。

よって EXPRESSの予約語に対応するJavaの予約語が何であるかを EXPRESS-G表記を持つものと持たないものに分けて調査する。その対応関係を表1に以下のように分けて示す。

- ◇ Java・Java-gともに対応 … ◎
- ◇ Javaのみ対応 … ○
- ◇ Java-gのみ対応 … △
- ◇ 代替機能 … ☆

◇ 宣言文も対応 … セル内塗りつぶし

◇ 宣言文は対応せず … セル内色無

ここでいう代替機能とは同じ、もしくは類似した構文要素に対応はさせられないが、機能面では同等な予約語が存在することを示している。また、Java のライブラリとして新規にクラス(新規API、と呼ぶ)を作成しなければならないものも有り、それらを作成することによって対応可能となるものを表 1 最右列に表記した。ただし、多重継承に関する下型型拘束、AND 及び ANDOR は Java では対応できない。

表 1 EXPRESS と Java の対応関係

	Java													
	パッケージ	クラス	インターフェース	配列	基本型	メンバ	フィールド	修飾子	extends節	implements節	import文	実行文	既存API	新規API作成
スキーマ	◎													
インタフェース仕様										◎				
エンティティ型(TYPE)	☆													◎
NUMBER														◎
REAL					◎									
INTEGER					◎									
STRING														◎
BINARY														◎
LOGICAL														◎
BOOLEAN					◎									
ENUMERATION													☆	◎
SELECT														◎
ARRAY				◎									☆	◎
LIST													☆	◎
SET													☆	◎
BAG													☆	◎
明示属性							◎							
省略可能明示属性							△						☆	
誘導属性							△							
逆属性							△							
ABSTRACT								◎						
サブタイプ														◎
ONEOF														◎
AND														
ANDOR														
関数							◎							
手続							◎							
規則													☆	
定数								○	○					
局所規則													☆	
局所変数													☆	
総称														◎
一般化集合体														◎
被集合体														◎

以上をまとめると、図4のようにある概念(Concept)を図式表現であるEXPRESS-GとJava-gで表現すると、それぞれに特有の一部分を除き、ほぼ同じ内容を記述することが可能であるといえる。そしてそれらを言語表現のEXPRESS及びJavaに置換える際には更に詳細な情報を記述可能になるので、情報量はEXPRESS、Javaともに増大するが、一部を除き(表1参照)その多くは共有することができる。しかし、EXPRESSはそれらを実装するためにC++への変換を行なうのでいくらかの情報損失が、Javaはそのまま実装可能であるので情報は失わない。だから、より共有部分の多いEXPRESS-G・Java-g間で表1の対応関係に基づき変換を行ない、EXPRESS-G→Java-g→Java(→Java)というルートによるモデル化が望ましいと言える。

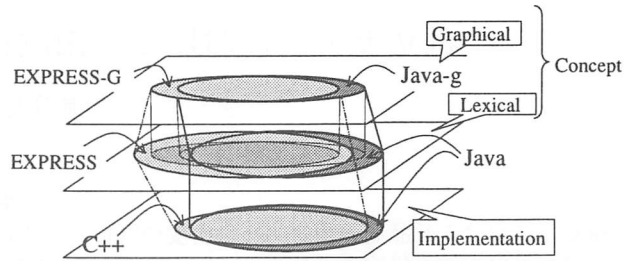


図4 EXPRESS と Java のモデル化可能範囲

4. 例題への適用

3. における対応結果の正当性を検証するため、例題として最も簡単な形状モデルを用いて、このモデルに関するEXPRESS記述からそのJava記述への置換え、及びEXPRESS-G表記からJava-g表記への置換えを行なった。図5にその図式表記法の対応の様子を示す。このようにほぼ同じような宣言形式で対応させ、同じ意味内容を表すことが可能である。

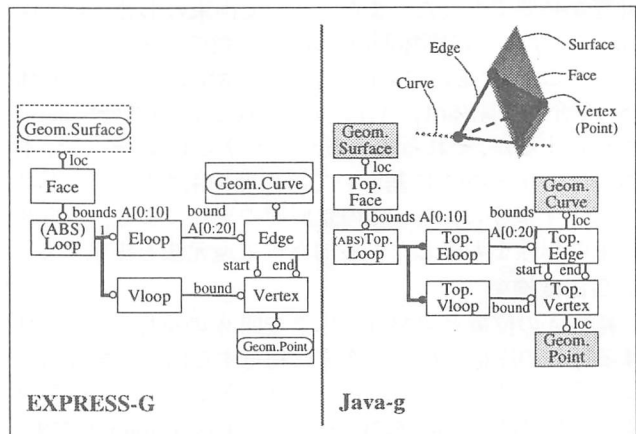


図5 簡単な形状モデル表現(左:EXPRESS-G、右:Java-g)

5. おわりに

本報では、以下の結論が得られた。

1. Javaは実装言語であるとともに形式言語としての役割を果たすことが可能である。
2. 図式表記法 EXPRESS-GとJava-gの表記法の対応関係を調査し、Java-g はEXPRESS-Gの持つ表記法をほぼ包含する。

参考文献

- [1] ISO Industrial automation systems and integration - Product data representation and exchange - Part11, 1994
- [2] James Gosling 他 : The Java™ Language Specification、Addison Wesley, 1996
- [3] Kari Kaitanen : jg book (Version 1.93)、VTT (<http://www.vtt.fi/cic/java/java-g/jg.htm>)、1997.12.16