

実行可能な形式仕様言語 CafeOBJ(3) CafeOBJ によるオブジェクト指向システムの 仕様記述ライブラリの記述*

本間毅寛^{†,1} 中川中^{†,‡} 澤田寿実^{‡,§} 谷津弘一^{†,2} 二木厚吉[§]

[†]情報処理振興事業協会 (IPA) 技術センター

[‡](株)SRA ソフトウェア工学研究所

[§]北陸先端科学技術大学院大学

1 はじめに

情報処理振興事業協会 (IPA) の実行可能な形式仕様言語の研究開発プロジェクトでは、実用的な形式仕様言語システムの基幹言語として、CafeOBJ を開発中である。CafeOBJ は代表的な代数仕様言語 OBJ に

- 動的なシステムの進化を記述するための書き換え規則
- 属性が変化するオブジェクトを宣言するためのクラス構文

を導入し、オブジェクト指向パラダイムの意味でのオブジェクトとその動的な振舞いを陽に記述できるようにした言語である。

従来の代数仕様言語の普及の障害として、ライブラリの不備と記述例の不足が挙げられている。CafeOBJ の基となっている OBJ のライブラリは基本的なデータ型のみからなり、現実のソフトウェアの仕様記述を効率的に行なうには不十分といえる。CafeOBJ の普及のためには、充実したライブラリと豊富な記述例は必要不可欠である。そのため、本プロジェクトでは CafeOBJ の開発と並行して

1. ライブラリの作成
2. 仕様記述実験による記述例の収集

*An executable specification language CafeOBJ(3)-CafeOBJ library for specification of a object-oriented system: Takehiro Honma, Ataru Nakagawa, Toshimi Sawada, Hirokazu Yatsu, Kokichi Futatsugi

[†]Information-Technology Promotion Agency, Japan (IPA) Software Technology Center

[‡]Software Research Associates, Inc. Software Engineering Laboratory

[§]Japan Advanced Institute of Science and Technology, Hokuriku

¹(株)SRA より出向中

²日本ユニシス (株) より出向中

を通して、CafeOBJ 普及に必要な環境の整備を行なっている。

本稿では、環境整備活動のうちの CafeOBJ ライブラリ作成の方針と現状について述べ、ライブラリ構成要素の具体例を紹介する。

2 CafeOBJ ライブラリ作成の方針と現状

現在までのところ、OBJ、Z、Larch といった形式仕様言語や ML 等のプログラミング言語のライブラリを参考にして、以下の方針に基づき仕様記述に必要なモジュールを CafeOBJ ライブラリの構成要素として作成してきた。

- 形式仕様記述言語全般による記述例からライブラリに必要なと思われるデータ構造を抽出する。
- プログラミングによく用いられるデータ型を抽象化する。
- 仕様記述実験からのフィードバックに基き、ライブラリの洗練と新たな構成要素の追加を行なう。

現在、ライブラリには、Z の数学的ツールキット相当のモジュールや、リストや待ち行列などの一般的なデータ構造を表現するモジュールがそろっている。

近來ソフトウェア開発におけるオブジェクト指向パラダイムの重要性が認識されている。前節でも述べたように、CafeOBJ はオブジェクト指向パラダイムをサポートする。CafeOBJ ライブラリには、オブジェクト指向パラダイムに基きシステムの仕様を記述するためのモジュール群も用意する。

3 ライブラリの具体例

ここでは、基本的なデータ構造であるが実際のソフトウェア開発に頻繁に用いられている、待ち行列の記

述を紹介する。

```

1. module QUEUE[X::TRIV]
   using INT .
2.   sort Queue, NeQueue .
3.   subsort Queue > NeQueue > Elt .
4.   op empty : -> Queue .
5.   op append_ : Elt Queue -> NeQueue .
6.   op head_ : NeQueue -> Elt .
7.   op tail_ : NeQueue -> Queue .
8.   op len_ : Queue -> Int .
9.   var E, E1 : Elt, Q : NeQueue .
10.  eq head (append(E,empty)) = E .
11.  eq head (append(E,Q)) = head (Q) .
12.  eq tail (append(E,empty)) = empty .
13.  eq tail (append(E,Q)) = append(E,tail(Q)) .
14.  eq len (empty) = 0 .
15.  eq len (append(E,Q)) = len(Q) + 1 .
16. endm

```

ここで、TRIV は次のようなソートの一つだけ持つモジュールである。

```
module TRIV is sort Elt . endm
```

INT は、整数とそれに作用する演算群を表すモジュールである。モジュール QUEUE はパラメータ付きモジュールであり、仮引数 X にモジュールをバインドさせて使用する。バインドされたモジュールは TRIV の制約 (ソートが 1 つ存在すること) を受ける。上の例では、3 行目で、2 行目で宣言したソート間の包含関係を宣言している。4-8 行目は待ち行列に作用する演算とその型の宣言であり、演算間に成り立つ関係は 10-15 行目の等式により定義されている。

次の例は書き換え規則とクラス構文を用い、待ち行列とその上の操作をオブジェクトと状態の変化として記述したものである。

```

1. module DynamicQUEUE[X::TRIV]
2.   using QUEUE[X] .
3.   class DQueue | queue : Queue .
4.   op Add_ : Elt DQueueId -> Message .
5.   op Delete_ : DQueueId -> Message .
6.   op Top_ : DQueueId -> Message .
7.   op Size_ : DQueueId -> Message .
8.   op _'s-top-is_ : DQueueId Elt -> Message .
9.   op _'s-size-is_ : DQueueId Int -> Message .
10.  var Q : Queue, NeQ : NeQueue, E : Elt .
    var DQ : DQueueId .
11.  rl Add E DQ
    <DQ : DQueueId | queue : Q> =>
    <DQ : DQueueId | queue : append E Q> .
12.  rl Delete DQueueId
    <DQ : DQueueId | queue : NeQ> =>
    <DQ : DQueueId | queue : tail NeQ> .
13.  rl Top DQ <DQ : DQueueId | NeQ> =>
    <DQ : DQueueId | NeQ>
    DQ 's-top-is (head NeQ) .
14.  eq Size DQueueId <DQueueId:DQueue | Q> =>
    <DQ : DQueueId | NeQ>
    DQ 's-size-is (len Q) .
15. endm

```

CafeOBJ では、オブジェクト指向システムの状態をメッセージとオブジェクトの混在したもので表す。上の例では、3 行目では待ち行列を属性に持つオブジェクトのクラス DQueue を宣言している。クラス DQueue のオブジェクトはソート DQueueId のユニークな識別子を持つ。4-7 行目ではオブジェクトに送られるメッセージを構成する演算を定義しており、9、10 行目の書き換え規則ではメッセージを受けたオブジェクトがその状態を変化させる様を表している。また、13、14 行目の書き換え規則ではオブジェクトは自身の状態を変えずにメッセージを出力する。メッセージで指定された識別子と同じ識別子を持つオブジェクトに対してのみ、これらの書き換え規則は適用される。

4 おわりに

実行可能性については様々な危険性が指摘されているが、プログラミング言語の修得が自作のプログラムを動かすことから始まるように、形式仕様言語においても自作の仕様を実行できることが作成/理解の助けとなる。ライブラリの作成においても、CafeOBJ の実行可能性は大いに役立っている。実行可能性は形式仕様言語の普及ということを考えると望ましい条件であるといえるのではないだろうか。

ライブラリは仕様を記述するにあたっての教科書となるべきものであり、この意味からも形式仕様言語の普及に必要な不可欠なものであると言える。さらなる CafeOBJ ライブラリの充実が望まれる。

謝辞 本稿で報告した試みは情報処理振興事業協会技術センターにおける実行可能な形式仕様言語システムの研究開発プロジェクトで行われた。

参考文献

- [1] K.Futatsugi and T.Sawada, A Preliminary View of CafeOBJ: a multi-paradigm language of Cafe environment, internal memo, 1992.
- [2] J.Goguen, T.Winkler, J.Meseguer, K.Futatsugi and J.-P.Jouannaud, Introducing OBJ, Technical Report SRI-CSL-92-03, Computer Science Laboratory, SRI International, 1992. To appear in J.A.Goguen editor, Application of Algebraic Specification using OBJ, Cambridge University Press.
- [3] J.J.Horning, J.V.Gutttag, Larch: Languages and Tools for Formal Specification, 1993.
- [4] J.M.Spivey, The Z Notation: A Reference Manual 2nd ed., Prentice Hall, 1992.
- [5] 谷津、本間、中川、二木, 実行可能な形式仕様言語 CafeOBJ の概要, IPA 第 12 回技術発表会論文集, 1993.
- [6] 本間、中川, 代数的仕様記述言語 CafeOBJ, 第 4 回情報処理学会全国大会講演論文集, 1993.
- [7] CafeOBJ 言語仕様書, 1993.