

オブジェクト指向モデル記述言語のインタプリタ処理系の開発

7N-7

中野喜之 小飼敬 金村星吉 高橋大輔 上田賀一

茨城大学 工学部 情報工学科

1 はじめに

本研究室ではメタ階層に基づくプロトタイプ方式によるモデルベースソフトウェア開発基盤の開発を進めてきている [1][2]。この開発基盤は図1に示すメタ階層構造を持ち、各階層を各レベルのユーザが受け持つ形態を取る。これまで記述言語に用いてきたC++がコンパイラ言語であるため、モデルの動的記述の実現が困難であった。そこで、開発基盤のメタモデルが有する概念を損なわずに記述・解釈できる記述言語としてオブジェクト指向インタプリタ言語 Bramble を開発することにした。この言語は、本研究で提供する部分、即ち記述言語部分だけでなくメタモデル部分も包含したものとなる。本報告では、言語の特徴と基底オブジェクト、基本動作について説明する。

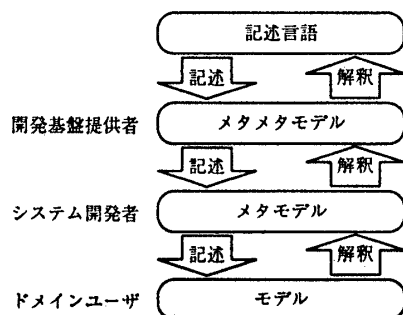


図1: モデルのメタ階層構造

2 言語の特徴

本言語の特徴を以下に挙げる。

- 全ての要素をオブジェクトとして扱う。

オブジェクトの集合は、その集合で一つのオブジェクトとして定義される。また、言語自身が振舞うために必要な要素もオブジェクトとして持つ。この全てのオブジェクトの基底となるオブジェクトをベースオブジェクトと呼ぶ。言語はこのベースオブジェクトから構成されている。

- コピー操作を主体としたオブジェクト指向の概念を用いている。

これによってクラス-インスタンスの概念が無くなる。しかし、ユーザの扱い方によって、オブジェクトをクラス-インスタンスのように振舞わせることができる。

- オブジェクトは属性だけを含む。

オブジェクトは変数とメソッドを区別せず、属性として包含する。

- メソッドをオブジェクトとする。

メソッドの役割を持ったオブジェクトをメソッドオブジェクトと呼び、メソッドオブジェクトがオブジェクトの属性となる時、この属性をメソッド属性と呼ぶ。

本言語と他のオブジェクト指向言語 [3]-[6] との比較を表1に示す。

3 オブジェクト

3.1 ベースオブジェクト

ベースオブジェクトは、本言語の基底となるオブジェクトであり、本言語は、このベースオブジェクトから構成されている。また、ベースオブジェクトは、ユーザが直接変更することはできない。ベースオブジェクトの種類は Object, String, True, False, Template, NULL, Widget, Database となる。

3.2 オブジェクトの関係

全てのオブジェクトは、所有、継承という関係で結ばれている。オブジェクトの所有関係は、既定義オブジェクト root からのツリー構造またはグラフ構造として表される。root は、所有関係の構造において根となるオブジェクトである。この時ベースオブジェクトは、root に所有され、root の属性となる。また、ユーザが記述したプログラムは、root のメソッド属性として定義され、本言語に実行される。

存在する全てのオブジェクトは root から到達可能であり、到達不可能なオブジェクトは、存在しないオブジェクトとなる。

オブジェクトの継承関係は、スーパーサブ関係の有向グラフで表され、存在するオブジェクト毎にある。

表 1: Bramble と他言語の比較

比較内容	Bramble	Smalltalk	CLOS	Eiffel	C++
class-base/copy-base 型付け	copy-base 無	class-base 弱	class-base 弱	class-base 強	class-base 強
Interpreter/Compiler	Interpreter	Interpreter	Interpreter	Compiler	Compiler
オブジェクトの継承	動的	静的	動的	静的	静的
多重継承のサポート	有	無	有	有	有
GUI 構築支援機能の提供	有	有	無	無	無
Database I/F の機能の提供	有	無	無	無	無

3.3 オブジェクトの生成

オブジェクトの生成は、オブジェクトのコピーによって行なわれる。あるオブジェクトから、異なる別のオブジェクトを生成するには、そのオブジェクトをコピーし、属性を追加または変更する方法と、Template オブジェクトをコピーし、拡張しようとするオブジェクトを継承した上で、オブジェクトに属性を追加する方法の二種類がある。この方法は、ユーザが目的によって選択する。

3.4 属性の操作

オブジェクトの属性に対しては、追加、変更、削除、検索の操作が行なえる。

属性の追加は、オブジェクトに新しく属性を追加し、その属性の中身をもったオブジェクトを属性に参照させる。この操作によって、属性を追加したオブジェクトと、属性が参照しているオブジェクトの間に所有関係ができる。

属性を削除することで、その属性が参照しているオブジェクトとの所有関係が切り離されることになる。この属性削除の操作によって、オブジェクトは削除される。

4 基本動作

本言語では、メッセージ送信が基本動作となる。あるオブジェクトが、あるオブジェクトに対してメッセージを送り、その結果を得ることができる。実際にメッセージを送信する時は、

オブジェクト メッセージ名 メッセージ引数群
という形式をとる。ここで、メッセージ引数群は一つ以上のメッセージ引数の並びであり、メッセージ引数はメソッド属性の実行時に必要となる引数で、

タグ名: オブジェクト
の組を一つの引数とする。このメッセージの集合が、オブジェクトのメソッドとなる。

メッセージを受け取ったオブジェクトは、それに対応したメソッド属性を評価する。評価されるメソッド

属性は、文字列の値を持っていないなければならない。メソッド属性の評価は、上記のようなメッセージ形式のメッセージ送信の単位を作り、そのメッセージを送信することである。

このメッセージ送信の操作を繰り返すことが、プログラムの実行となる。

5 おわりに

モデルベースソフトウェア開発基盤の記述言語を実現するためにオブジェクト指向言語 Bramble を開発した。本言語は、コピーベースのオブジェクト指向の概念を導入したインタプリタ言語であるため、モデルの動的な記述を容易にすることができる。今後、本言語の目標として、エディタ、デバッガなどのプログラミング支援ツールを持たせ、開発効率の良いモデル記述環境を構築することが挙げられる。

参考文献

- [1] 上田賀一, 安間その美, 高橋大輔: “メタ階層に基づくモデルベースソフトウェア開発基盤の提案”, ソフトウェア工学の基礎 II, pp.11-20, 近代科学社 (1996)
- [2] 安間その美, 高橋大輔, 上田賀一: “システムプロトタイプリング支援のための基盤機構の開発”, 情報処理学会ワークショップ論文集, Vol.95, No.1, pp.49-56 (1995)
- [3] Bertrand. Meyer (二木厚吉監訳): “オブジェクト指向入門”, アスキー (1990)
- [4] L. J. Pinson, R. S. Wiener (富士ゼロックス情報システム訳): “Smalltalk: オブジェクト指向プログラミング”, トッパン情報科学シリーズ 11 (1993)
- [5] 所真理雄, 松岡聡, 垂水浩幸: “オブジェクト指向コンピューティング”, 岩波コンピュータサイエンス (1993)
- [6] 本位田真一, 山城明宏: “オブジェクト指向システム開発”, 日経 BP 社 (1993)