

2M-2 オブジェクト指向知識ベース管理システムJasmine におけるコンパイラ

青島正明 鴻江美智子 小櫻文彦 鈴木文雄 山根康男 石川博
(株)富士通研究所

1.はじめに

CADや知識処理の分野ではマルチメディアや知識といった複雑なデータを高速に格納・管理することが要求されている。そこで、我々はマルチメディア・データや知識を高速に処理し、格納・管理できるオブジェクト指向知識ベース管理システムJasmineの研究開発を行なっている。

Jasmineはオブジェクト指向言語と従来の関係データベース操作言語の融合を目標に設計した知識ベース操作言語Jasmine/C[1]を提供し、我々はそのJasmine/Cを高速に実行することを目的にJasmine/Cコンパイラとインタラクティブな環境で使うためにインタプリタを開発した。本稿では、単一のオブジェクトを操作するメッセージ送信式やオブジェクトの集合を検索する条件検索文の実行時における知識ベース上の辞書情報、デモン、メソッドへのアクセスを減らすために、それらの情報を生成コードに展開するコンパイル方式について述べる。

2.Jasmine/Cの特徴

Jasmine/Cは次に示す特徴をもち、アプリケーションの開発を容易にしている。

- 1)データ間の関係を陽に表すこと、属性値として集合を許すことなどにより、対象領域のモデルを自然にまた容易に知識ベース化することができる。
- 2)知識ベース上のデータやマルチメディア・データを簡単に検索できる。特にドット記法の導入により、データ間の関係を容易にたどることができる。
- 3)外部記憶上のデータを主記憶上のデータと全く同様に扱える。
- 4)データにアクセスしたときに付随的に起動されるデモンにより、データ間の一貫性を保つことができる。
- 5)クラスにより変数の型を指定したとき、その変数は、指定したクラスのインスタンスだけでなく、その下位クラスのインスタンスも含まれる。

3.Jasmine/Cコンパイラのシステム構成

図.1にJasmine/Cコンパイラのシステム構成図を示

す。Jasmine/Cコンパイラは、Jasmine/C言語で書かれたプログラムをコンパイルし、コードとしてC言語を生成する。Jasmine/Cコンパイラは、字句・構文解析部、意味解析部、最適化部、コード生成部から成り、意味解析部と最適化部において、オブジェクト管理部を通して、知識ベース上のメソッド、デモン、辞書情報を参照する。次にコード生成部においてそれらを直接コード上に展開する。オブジェクト管理部は、オブジェクトの生成や参照や更新を行ったり、オブジェクトの集合を操作するライブラリからなる。その下にレコードレベルの参照・更新等を行ったり、関係演算を行ったりする拡張データベースエンジンをもつ。

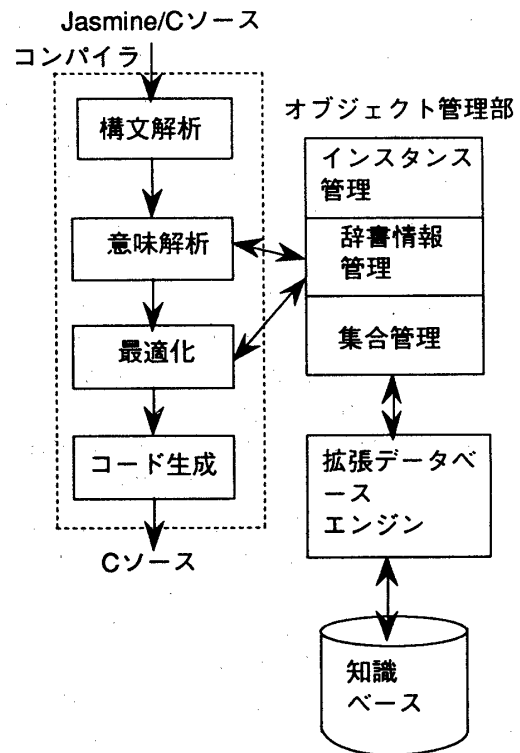


図1 Jasmine/Cコンパイラのシステム構成図

The Compiler of An Object-Oriented Knowledge Base Management System Jasmine
Masaaki Aoshima
FUJITSU LABORATORIES, Ltd.

Jamine/Cコンパイラが生成するコードは、オブジェクト管理部の中で、オブジェクトの基本的な動作を行なう関数の呼び出し形式である。したがって、オブジェクトの属性値を参照したり、更新したり、オブジェクトを生成したりする動作以外の知識ベースへのアクセスを含む機能(メソッド、デモン)については、意味解析や最適化時に知識ベース上にアクセスし、生成コード上に展開する。また、オブジェクト管理部は、これらの基本的なオブジェクト操作を行なう関数インタフェースの他に、知識ベース上のメソッド、デモン、辞書情報へのアクセスを含むインタフェースをもつ。最適化のオプション指定によって、このレベルのインタフェースを生成し、生成コード量を増やさないうコードも生成することもできる。さらに、C言語の中に、これらの関数インタフェースを直接書くことによってもJasmine/Cプログラムと同等の機能を実現することができる。集合操作を含む条件検索文に関しては、最適化部において、選択演算、結合演算、unnest演算(多値属性が指定されているときそれを1NFにする演算)、nest演算(多値属性が指定されているときに、それらを正規化する演算)等の組み合わせに変換し、それらの関数呼び出しの並びにコード生成される。条件検索文を直接文字列として受け取り、それを実行する関数インタフェースがある。

4. Jasmine/Cコンパイラの方式

本システムのメッセージ送信式は、知識ベース上にあるオブジェクトにメッセージを送信し、そのメッセージに関連する知識ベース上にあるメソッド、デモン、辞書情報を検索し、それを実行するという処理を含む。そのため、実行時における知識ベースへのアクセスが頻繁に起こり、実行速度が遅くなるということが考えられた。そこで、オブジェクト管理部の中の基本的な動作を行なうインタフェースをコード生成したり、ユーザ定義のメソッドに関しては、それに対応するCの関数の呼び出し形式を生成する。ユーザ定義のメソッドに対応するCの関数は、生成コードとオブジェクト管理部のライブラリがリンクされるときに結合される。

次に、メッセージ送信式の実行時における知識ベースアクセスを減らす具体的な方式を示す。

1)メソッド、デモンの検索

メソッドやデモンは、システム定義の手続きクラスのインスタンスとして知識ベース上に登録されて

いる。本言語では、ユーザ定義のメソッドには、前置デモン、後置デモンを付加したり、静的属性の操作には、未定値デモン、制約デモン、更新デモン、追加デモン、削除デモンを付加したりでき、さらに下位クラスに継承される。これらのものを実行時に知識ベース上に探索していくとき、インヘリタンスを解決していると多大なオーバーヘッドがある。そこで、これらのメソッド、デモンは対応するCの関数を直接呼び出すコードを生成している。

2)静的属性に関する操作

静的属性の参照、更新、追加、削除は、本システムの基本的な動作であり、頻繁に利用される。そこで、これらに関しては、送り先オブジェクトのOIDと静的属性のレコードにおけるフィールド番号、静的属性の型、静的属性の値などのパラメータをもつオブジェクト管理部の基本関数インタフェースをコード生成することによって、送り先オブジェクトが属するクラスのもつスキーマ情報へのアクセスを減らしている。

3)下方探索

あるクラスを型として指定したオブジェクト変数は、そのクラスに下位クラスが存在するときには、それ自身のクラスとその下位クラスを含む変域をもつ。メソッド、ファセットを探索するとき、それらの複数のクラスにわたって探索する必要がある。これを下方探索と呼ぶ。

このとき、一般に上位のクラスで定義されたメソッド、ファセットが、下位クラスにおいてオーバーライドされているとき、コンパイル時に、メッセージに対応するメソッドやデモンを決定できない。そこで、コンパイル時に、指定されたクラスと、その下位クラスにおいてメソッド、ファセットをオーバーライドしているクラスをあらかじめ探索する。送り先オブジェクトが属するクラスを判定し、メソッドを実行するif文の並びにコードを生成する。

5.おわりに

Jasmine/Cコンパイラのシステム構成について述べ、実行時における知識ベース上のメソッド、デモン、辞書情報へのアクセスを減らし、Jasmine/Cを高速に実行する方式について述べた。

[参考文献]

- [1] 青島,泉田,牧之内,鈴木,山根 The C-based Database Programming Language Jasmine/C, Proc. 16th VLDB Conference 1990