

1E-2

対象指向型言語をベースにした
多言語モジュラープログラミングの提案

中所 武司 (日立製作所システム開発研究所)

実世界のマクロな計算モデルの表現に適した対象指向型言語と詳細な計算手順の表現に適した従来言語を融合した多言語モジュラープログラミングMMP-85を提案する。モジュール間の静的関係を階層構造、動的関係をメッセージ伝送で表現する対象指向型言語をベースにし、モジュールの内部は従来言語で記述することにより、個々のモジュール開発には従来の財産を生かしながら、全体的モジュール間関係は実世界に近い記述を行うような大規模ソフトウェア生産方式を可能とする。

1. はじめに

大規模ソフトウェアの生産技術の中核をなす設計・プログラミング技法の主目的は、対象とする問題(応用分野)に適した計算モデルの構築とそれに対応したプログラムモジュールの構成であり、70年代以降、多くの設計技法やプログラミング技法が提案されてきた。

しかしながら、それらの技法の実現手段であるべきプログラミング言語は、50年代、60年代に開発されたFortran, Cobol, PL/Iなどが主流を占めている。即ち、フォンノイマン型計算機アーキテクチャの計算モデルの記述に適した従来の手続き型言語を用いて、人間世界の計算モデルを記述しているわけである。

われわれは、6年前にその1つの解決策として、モジュール間関係とモジュール自身の記述を分離し、計算モデルはモジュール間関係記述言語で記述すると共に、個々のモジュールの記述は、その機能に適した言語を自由に選べるようなプログラミング方法論を提案し、多言語モジュラープログラミング(Multilingual modular Programming, 以下MMP-79と略す) [1]と名付けた。

本論文では、対象とする問題をより自然に表現する技法として注目されている対象指向型言語 [4] をモ

ジュール間関係記述言語として位置づけることにより、MMP-79の新たな実現方法MMP-85を提案する。

2. MMP-79の概要

MMPの主目的は、次の2点である。

- (1) モジュール間関係をインターフェイスも含めて正確に記述できること
- (2) 各モジュールの機能に合った記述が可能であること

その実現方式としては、(1)については、プログラミング言語の機能として取り込むよりも、独立した言語とすること、また(2)については、Adaのように必要な機能を単一言語で実現する巨大化アプローチよりも複数の簡潔な言語を用意することを基本方針とした。詳細は文献 [1] に詳しい。

3. MMP-85における対象指向型言語の位置づけ

対象指向型言語におけるオブジェクトをMMPにおけるモジュールとみなすことにより、対象指向型言語をMMPのモジュール間関係記述言語として位置づけることができる。即ち、クラスオブジェクトの階層構造と性質の継承規則が静的関係を表現し、インスタ

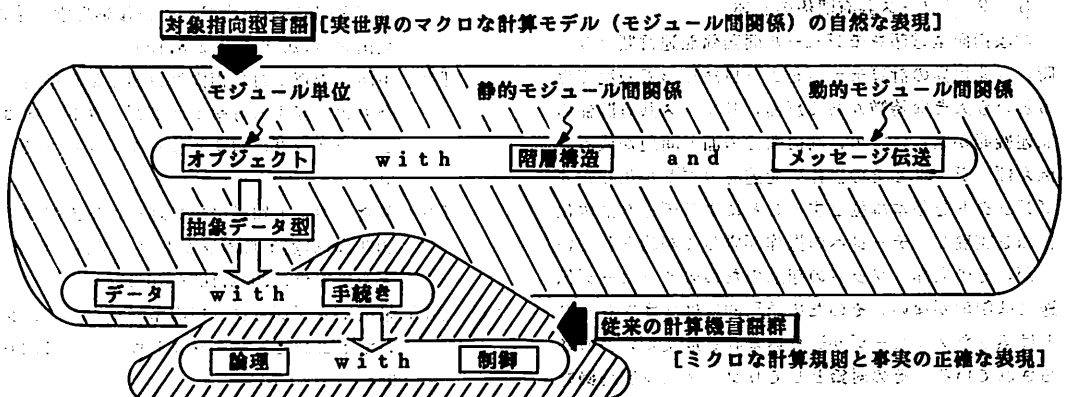


図. 多言語モジュラープログラミングMMP-85の設計思想

オブジェクトの生成とメッセージ伝送が動的関係を表現する。一方、個々のオブジェクト内の手続きは従来の言語で記述できるようにする(図参照)。本章では、このようなMMP-85を具体化するための方式について述べる。

3.1 階層構造と継承規則の解釈

対象指向型言語の枠組みの中で、各クラスの定義を従来の言語で行う方式において、従来の言語の意味を変えずに済むかという問題がある。それは、言い換えれば、クラスの階層構造における継承規則をどのように解釈するかということである。

この継承規則の解釈には、次の2通りがある。

(1) コピールール:

インスタンスの場合、その元のクラスおよびその上位のクラスのインスタンス変数とインスタンスメソッドがすべてそのインスタンス内にコピーされたと考える。

(2) スネカジリルール:

メッセージを受信したクラスあるいはメッセージを受信したメソッドの定義がなければ、自分の親クラスにそのメッセージの処理を依頼する。

以上の2方式は、対象指向型言語の世界では同じ意味と考えられるが、クラス定義の動的変更を許す場合は、2方式の間に意味解釈の相違が生じる可能性がある。

MMP-85では、継承規則の解釈は、モジュール記述言語として従来の言語が複数用いられ、かつ上下関係のあるモジュール(オブジェクト)が異なる言語で記述された場合はスネカジリルールが適する。一方、従来の言語が1つ、あるいは上下関係のあるグループ内では1つだけに限定するならば、コピールールの方が、オブジェクト単位に従来の言語の閉じた世界を形成できるため、適当である[2]。

3.2 リソース共有関係の記述

対象指向型言語をモジュール間関係記述言語とみた時、人間世界の計算モデルをより自然な形で表現できるという大きな特徴がある反面、リソース共有関係の記述機能が乏しいため、高信頼性の面で問題が残る。例えば、データとその操作の記述をカプセル化したオブジェクトを基本単位とすることによりデータ抽象化技法を実現し、データへの操作が厳しく制約されているが、その操作メソッドの起動のためのメッセージ伝送には制約がない。そのため結果的には各オブジェクトは誤ったメッセージが送り込まれる危険にいつもさらされている。その対策としてリソース共有制約機能を導入する。

3.3 異種言語間リンケージ法

- (1) モジュール間の実行制御はメッセージ伝送によって行う。この時、返答の仕方を幾種類か用意することにより並行処理、逐次処理をはじめ、種々の制御方式が可能となる。返答の仕方としては、返答は別のメッセージ伝送とするかあるいは同一通信路と考えるか、また返答の相手は送り手に限るかあるいは任意のオブジェクトを指定できるか、あるいは送り手は返答の到着まで待つか否か、待つとすればメッセージ伝送直後で待つかそれともメッセージ受信可能状態で待つか、等がある。
- (2) パラメータメカニズムはメッセージ伝送方式との対応でcall by value方式、即ち値渡しを基本とする。更に(1)の返答方法と絡むが、結果の値を戻す場合はcall by value and/or result方式に限定する。但し、値としてポインタ値を許したり、プログラムを値として渡したりするような拡張を行えば、かなりの柔軟性が得られる。
- (3) データ型の不一致については、データ型変換を何らかの方法で行う必要があるが、ここでは、メッセージの受信時および計算した結果の返送時に受信側でデータ変換デモンを起動するものとする。このデモンは必要ならばユーザ側で定義可能とする。デモン起動は送信側で行うことも考えられるが処理が分散される。返す値の変換を受信側で行うためには、送信側で型情報を渡せばよい。

4. おわりに

現在、対象指向型言語をベースに、モジュール記述言語として論理型言語を採用した知識情報処理記述言語Super-LONLIを開発中である。[2, 3]

参考文献

- [1] 中所: 多言語モジュラープログラミングとその処理系LIGERの概念設計: 情報処理学会第21回全国大会, 2C-6, 287-288, 1980.
- [2] 中所, 芳賀: 対象指向型言語と論理型言語の融合方式に関する考察: オブジェクト指向コンピューティングに関するワークショップW00C, 1985.
- [3] 芳賀, 中所: 知識情報処理システム記述言語S-LONLIの提案: 日本ソフトウェア科学会第1回大会論文集, 2D-2, 167-170, 1984.
- [4] 米沢: オブジェクト指向型プログラミングについて: コンピュータソフトウェア, 1, 1, 29-41, 1984.