

## 特集「プログラミング言語最新情報」の編集にあたって

松田 裕幸†

本特集では最近のプログラミング言語として、1980年代以降に生まれたオブジェクト指向言語 Eiffel, C++, 生まれは70年代と古いが一般にはあまり知られていない関数型言語 ML, Scheme, さらに読者にとってすでになじみがありその最近の動向が注目される Fortran, COBOL および Ada について取りあげる。

本特集にあたりどの言語を取りあげるかの明確な基準を持っていたわけでもなく、学会誌編集委員各自のバックグラウンドによる偏りも否定できない。昨年開かれた第2回 HOPL (History of Programming Languages Conference, ACM SIGPLAN Vol. 28, No. 3, 1993) で歴史的に価値ある言語として、「1982年までに公にされ、85年までに使用もしくは教授されていること」という基準により選ばれたプログラミング言語は以下の通りである：Ada, ALGOL 68, C, C++, CLU, Concurrent Pascal, FORMAC, Forth, Icon, Lisp, Pascal, Prolog, Smalltalk, 各種シミュレーション言語。これに対し、学会誌編集委員会が候補に上げた各プログラミング言語が、結果として、日本におけるプログラミングの「現場」を反映し、その将来に影響を与えるものになっていることを願う。当然読者からのご意見、ご批判を待ちたい。

言語の選択に関しては上記のような状況ではあったが、各記事の構成に関しては明確な方針を最初から持って執筆依頼を行った。一つは、言語仕様や特徴を単に網羅的に列挙するのではなく、執筆者自身がその言語の鍵と判断した特徴を重点的に解説する。ある意味では執筆者の主観的側面を許すもので、これは各記事の副題に反映されている。たとえば、Scheme の場合には「抽象化能力をもつ goto-less 手続き型言語」という副題がついている。多少の議論は覚悟した上での表現である。二つ目は、各種言語的特徴がソフトウェア方法論との関連でどう見えるかを述べる。例えばモジュール機能は方法論との対比では情報隠蔽をサポートする。あるいは、ML に代表される自動型推論機

構はソフトウェアの信頼性を高めるものである。標準化動向、各言語に関する情報の入手方法、特にインターネットを含む、なるべく紹介する点にも配慮した。以下、各記事の構成を述べる。

C++, Eiffel は一般にオブジェクト指向の枠組でくられるが、前者がCを前提とし、厳密な意味でのオブジェクト指向モデルを議論するよりは実用性を重視しているのに対し、後者はソフトウェア設計論との関連を強く意識し、プログラム仕様言語としての立場を強調している。ML と Scheme は代入を許す代表的な関数型言語である。後者が型に関しほとんど考慮していないのに対し、前者は型を要請する。しかしながら ML は型に関し、推論する能力 (Pascal のように明示的に宣言しなくても型に関する推論規則を用いることで型を判断) を持つことで、代入機能と相まって、信頼性を保証しつつ高い実用性が望める言語として有望視されている。一方、Scheme はここでは、むしろ、データ抽象および制御抽象の高い手続き型言語という立場からながめている。

Ada, COBOL では共に標準言語仕様としての「オブジェクト指向」が紹介される。特に Ada は 1983年 ISO 規格になってから10年日を経て、今日仕様改定作業が行われており、タスク間相互排除(モニタ風)、ライブラリの階層化等の各種機能の追加が予定されている。一方、COBOL に関しては、1997年に予定されている ISO COBOL 97 の概要や、各国語文字機能、整構造プログラミングなどについて紹介し、また、30年以上におよぶ言語仕様の変遷についても述べる。Fortran は並列処理にポイントを置き、従来からあった共有メモリマシン用モデルに対し分散メモリマシンを意識したモデルに基づいて、forall ループを並列処理するアイデアを中心に解説する。なお本特集は編集側の都合により第1部(Eiffel, ML, Scheme)、第2部(C++, Ada, Fortran, COBOL)と2回に分けて掲載いたします。最後に、本特集にあたり快く記事を書いていただいた各執筆者ならびに読者に感謝申し上げます。(平成6年2月7日)

† 東京工業大学総合情報処理センター