

スクリプト言語 GavaOne の設計と実装

木 村 紀 介

最近のプログラミング言語について考えてみるに、C 言語をはじめとする手続き型言語が隆盛であり、関数プログラミング言語は一時の勢いを失っているように思える。関数プログラミング言語は、人工知能研究とともに発展してきたが、その後のマルチメディア技術の高度化に対する対応の遅れは、無視できないものがある。この原因の 1 つとして、スクリプト言語の基礎をなす「中間コードとデータの形式」が複雑化する情報化ニーズに対応できないことにあると思う。「中間コードとデータの形式」の代表として、Lisp 言語の S 式 (symbolic expression) がある。中間コードとデータの形式に S 式を採用し、Lisp 言語自身で中間コードを生成する機能 (reflection) は、真に優れている。しかし、S 式は、シンプルであるがゆえに、複雑な処理機能を記述するには、可読性 (理解のしやすさ) を犠牲にしている面がある。Lisp 登場時には M 式 (meta expression) で可読性を補っていたが、LISP 言語仕様の早い段階で、S 式を中間コードとデータの形式に採用し、M 式は忘れ去られた。今後の関数プログラミング言語の発展のためにも、処理内容をより分かりやすく表現できる M 式を、今一度見直す時期に来ていると筆者は考えている。筆者は、プログラミング記述能力を上げるため、LISP 言語の M 式を取り入れた「新しい中間コードの形式」を考案し、スクリプト言語 GavaOne を設計、実装した。本発表において、「新しい中間コードの形式」を解説し、さらに、使用例によって GavaOne 言語の概略を説明する。

The Design of Scripting Language GavaOne and the Implementing

TOSHISUKE KIMURA

The author contrived “the form of the new intermediate-code” by which the M-form (meta expression) of the LISP language was introduced to give a programming description capability, designed and implemented scripting language GavaOne. As for the this presentation, “the form of the new intermediate-code” is explained and moreover, the outline of the GavaOne language is described by the use example.