

発表概要

メタレベルの拡張に適したメタレベルの構成法

田中 哲[†] 渡部 卓雄[†]

メタレベルアーキテクチャを採用するプログラムでは、通常、メタレベルは対象とする適応領域 - ベースレベルの構造やその拡張・適応方式等 - に特化された形で設計される。メタレベル自体を、アプリケーションに依存した部分とそうでない部分に分離できれば、後者を再利用することによってメタレベル設計が容易になる。そのために、本稿では拡張が容易なメタレベルの設計をアプリケーションに依存せずに行なう方法について論じる。拡張を容易とするために必要なことは、安定で見通しが良い構造を導入し、その構造に従ってモジュールを組み上げていけるようにすることである。そこで、本稿ではメタレベルアーキテクチャ一般について成り立つ構造としてベースレベル記述の構文木を考え、その構造の上にモジュールを載せるという方法を提案する。ここで、この方法では一つの構文に複数のモジュールを相互作用なしに共存させることができ、また、モジュール間のインターフェースを変数の単一代入として定義することにより見通しの良いモジュール間通信を実現する。これにより、モジュールを容易に後から導入してメタレベルを拡張することが可能となる。この方法の適用例として、簡単な言語のインタプリタ、および LR パーザジェネレータを示す。特に後者は（インタプリタやコンパイラなどの）典型的な言語処理系とは異なる構造のソフトウェアであるが、我々の方法はこのようなものにも適用できることを示す。

A Structuring Method for Extensible Metalevels

AKIRA TANAKA[†] and TAKUO WATANABE[†]

Usually, a metalevel is designed for a specific application domain, especially when the metalevel is dedicated for extensibility because details of the extensibility are very application specific issues. The aim of this study is to extract characteristics common to all metalevel architectures and to establish a method for structuring a metalevel using the characteristics for extensibility of the metalevel. The methodology eases metalevel designs and its extensions because it is applicable to most application domains since they are not depending to specific domains. The metalevel should be constructed by multiple modules organized as stable and open structures. We propose that the methodology structuring a metalevel by baselevel syntax and sticking semantic modules to the syntactic elements. Baselevel syntax is common characteristics for all metalevels since they have a definition of the baselevel and it contains the syntax definition. Therefore the structuring method is applicable to most metalevels. The modules interact with single assignment attributes defined for each edge of baselevel syntax tree. Since the attributes can be defined as many as required, multiple semantic modules can be stucked to on syntactic element without any interference. Consequently we can introduce new modules for extensions easily. This paper explains the methodology by examples including lambda calculus and LR parser generator.

(平成 11 年 10 月 30 日発表)

[†] 北陸先端科学技術大学院大学情報科学研究科
School of Information Science, Japan Advanced Institute of Science and Technology