

Title	拡張の合成を容易とするメタレベルアーキテクチャの研究
Author(s)	田中, 哲
Citation	
Issue Date	2000-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/895
Rights	
Description	Supervisor:渡部 卓雄, 情報科学研究科, 博士

A study for extensible metalevel architectures

Akira Tanaka

School of Information Science,
Japan Advanced Institute of Science and Technology

February 17, 2000

Abstract

Full-featured and large scale software development needs a method that constructs extensible and adaptable software at a low cost. It is an effective method for keeping software kernel small and maintainable to make software extensible and adaptable. There are two important problems with that.

- A single extension should be easy to describe.
- Conflicts of multiple extensions should be detected and should be able to resolve.

The idea “metalevel architecture” is used as general solution to the former problem. The metalevel architecture separates software into a metalevel and a baselevel. A baselevel is a description which is interpreted by a metalevel. Metalevel extension modules based on the metalevel architecture provides flexibility of design for extension method. There are proposed systems along to the metalevel architecture: extensible interpreters, compilers, operating systems, etc.

But the latter problem is not solved by the metalevel architecture in general. Applying multiple extensions may not work well even though individual extensions work well when they are applied separately. Since an extension may be developed without a knowledge of another extension, it is very hard to check their conflicts. This is caused by that interaction between extensions are not clear.

This study proposes a metalevel construction method which defines the interaction more clearly using an attribute grammar: it defines a baselevel as an attributed tree and a metalevel extension module as a set of procedures which defines an attribute value. The extension modules interact each other by single-assigned attributes. When an additional extension module is applied, a conflict is detected reliably. The method can be applicable to various software not only language systems. This paper explains the method by (1) simple functional language corresponding to lambda calculus and (2) LR parser. (2) shows that the method can be applied other than language systems

Key Words: metalevel architecture, extension, conflict resolving