

Summary 5 Jianfeng Jia

Reflection and Semantics in LISP

This paper presented a general architecture called procedural reflection to support self-directed reasoning in a serial programming language. This architecture uses an infinite tower of meta-circular interpreters. A meta-circular evaluator is a special case of a self-interpreter in which the existing facilities of the parent interpreter are directly applied to the source code being interpreted, without any need for additional implementation. Each meta-circular interpreter executes the one under itself and the bottom one executing the end-user program. Reflective computations are initiated by calling reflective procedures, procedures with three parameters, the body of which being executed one level up in the tower; upon invocation, a reflective procedure is passed a reification of the argument structure of its call, its current environment and its current continuation. The characteristic of self-interpreters gives the LISP an advantage of implementing the reflection.

Aspect-Oriented Programming

In this paper they introduced the idea of Aspect in programming language. Even though most classes in an OO model will perform a single, specific function, they often share common, secondary requirements with other classes, and the code needed to perform the secondary functionality is often identical. But it's hard to modularize by OO model. Typically, an aspect is scattered or tangled as code, making it harder to understand and maintain. (like logging, profiling ...). Aspect-Oriented Programming allowing the programmer to express cross-cutting concerns, which is cohesive areas of functionality across the different classes, in stand-alone modules called aspects. It entails breaking down program logic into distinct parts. It includes programming methods and tools that support the modularization of concerns at the level of the source code. All the implementations have some crosscutting expressions that encapsulate each concern in one place. The join point defined when the function will be called, and the declarations defined how to implement the behaviour.