CodeMate

An automate tool for programming
and templating

by Li Dong (dongli@lasg.iap.ac.cn)

1 Introduction

CodeMate is an automate tool for processing and building project (currently only in Fortran).

1.1 Why CodeMate?

There are many programming languages for helping people to solve problems, and generally they fall into two categories, one is compiling languages and the other is interpreted languages. The major target language that CodeMate deals with is Fortran, which is the first high-level language and is mainly used in numerical computation due to its high efficiency.

(some more background?)

It is tedious and error-prone to manage the Makefile of a project. Sometimes, user just wants to write a small and quick project for proving concept, but writing the dependencies among the codes and linking with external libraries is such a headache. In the other hand, even a number of starting simple codes maybe group up into a huge project with hundreds of codes. In this case, the headache turns to be serious.

For most other languages, like C++ and Java, there are many great ingerated development environment (aka IDE) to do the dirty jobs. Fortran only gets old Compaq Visual Fortran, expensive Intel Visual Fortran, and several less used IDEs. Almost all of them are only availible in Windows, but most serious Fortran developers work in Linux or Mac. So CodeMate comes into business to close the gap for Fortran users (most of them are scientific programmers), and provides more capabilities that IDEs do not have. Also CodeMate is lightweight compared with IDEs, it is just one command at all (that is `codemate`), so it can be run on the remote server with only command line interface.

Using CodeMate, user will be liberated from writing Makefile or other duplicate stuffs. No file editing is needed for basic uses. In addition, CodeMate will do more thing automatically, such as linking external libraries automatically, and processing templates.

1.2 CodeMate Capabilities

CodeMate can scan a Fortran project to extract internal and external dependencies, and create a Makefile for it. If there are any template instance and CodeMate knows about the template definition, the template instance will be processed to generate a full code internally in `<project root>/.codemate/processed_codes`. After these operations, user can invoke `make` to build the project.

1.3 Template

Template, or more precisely template metaprogramming, is a technique to transform the original codes into temporary codes based on some rules. Many morden

programming languages support it natively (by compiler), such as C++. Unfortunately, Fortran does not support it formally. That means when you want to use some advanced data structure like linked list, you have to reinvent the wheels over and over again, due to the limited generic programming and the safe but also limited pointer. Fortran has its reason to insist these limitations, that is to ease the compiler optimization, but in real and modern applications, it is difficult and less maintainable to just use the basic data structure. CodeMate tries to solve this problem by constructing a practical template framework, which uses a customized Fortran parser[1], interface and dynamic compilation mechanism of Java. Do not be scared by Java, since normal users will not contact with it directly. Only advanced users who want to write templates need to learn some basic Java syntax. Once familiar with it, you will find out that you can create any useful templates.

2 Tutorial

2.1 Installation

2.2 Basic Usage

CodeMate tries to minimize the workflow of building a project, so it does. Only one operator (or subcommand) is needed normally, that is `scan`:

```
$ codemate scan <project root>/<single code>
```

When a directory is provided, CodeMate will consider it as the root of the project, and any Fortran code (with suffix as .F90 or .f90) will be processed. When only a single code is provided, the code will be parsed and print the rewrited code onto the console.

2.3 Template Usage

2.3.1 Write a Template

[1] The customized Fortran parser is generated by using ANTLR.