

資工系實務專題研究計畫成果報告

C++ Program Slicing (Eclipse Plugin)

專題編號：101-CSIE-S023

執行期限：100 年第 1 學期至 101 年第 1 學期

指導教授：劉建宏

專題參與人員： 98590326 潘其振
98590335 陳士民

一、摘要

隨著程式的規模不斷擴大，程式碼的複雜度亦隨之提升，為了讓 Programmer 在除錯、測試階段能夠有效率，我們利用 Program Slicing[1] 的技術，開發 C++ Program Slicing Eclipse Plugin，透過這個工具的輔助下，利用 Program Slicing 技術找尋 C++ 程式內部各變數之間的關聯，以達到讓 Programmer 快速搜尋程式碼錯誤部分的目的，並提升除錯的效率。

關鍵詞：Eclipse Plugin、Program Slicing

二、緣由與目的

測試是程式開發過程中重要環節之一，不可被忽略。隨著程式的規模不斷擴大，程式的複雜度亦隨之提升，在進行程式除錯、測試階段時，Programmer 常因大量且複雜的程式碼，在這個階段耗費過多的時間與精力。為了縮短 Programmer 在除錯、測試階段花費的時間，我們利用 Program Slicing 技術，開發 C++ Program Slicing Eclipse Plugin，利用這個工具先將 C++ 程式碼解析成 AST(Abstract Syntax Tree)，再使用 Visitor 尋訪各個節點，將相關節點取出來，並找尋相關節點對應之行數，最後將相關節點所對應的行數在 Editor 上標記出來，使用者可以快速對 C++ 程式進行 Forward and Backward 追蹤，找出相關的程式碼，讓使用者在 Eclipse 上對 C++ 程式除錯、測試時更加的快速方便。

三、使用技術和工具說明

(一) 使用工具

JDK(Java Development Kit)[2]
Eclipse IDE for Java Developers
Eclipse CDT(C/C++ Development Tooling)[3]
MinGW[4]

(二) 工具說明

我們的需求是做一個 Eclipse 的 Plugin，提供 Programmer 在 Eclipse 上開發 C++ 程式時使用，由於 Eclipse 本身是由 JAVA 開發而成，因此必須要在 JRE(Java Runtime Environment)[5] 下才能運行，而 JDK 包括了 JRE 以及開發過程中所需要的一些工具程式，因此在執行 Eclipse 之前必須先安裝 JDK。Eclipse 本身就提供開發 Plugin 的開發環境，因此可以直接在 Eclipse 上開發 Plugin。而 Eclipse 除了可以開發 Java 外，安裝 CDT 還可以開發 C/C++ 程式專案[6]，但 Eclipse 本身並無提供 C/C++ 的 Compiler，所以必須另外安裝 C/C++ Compiler，而我們使用的是 MinGW，如此一來便能使用 Eclipse 去開發 C/C++ 程式。

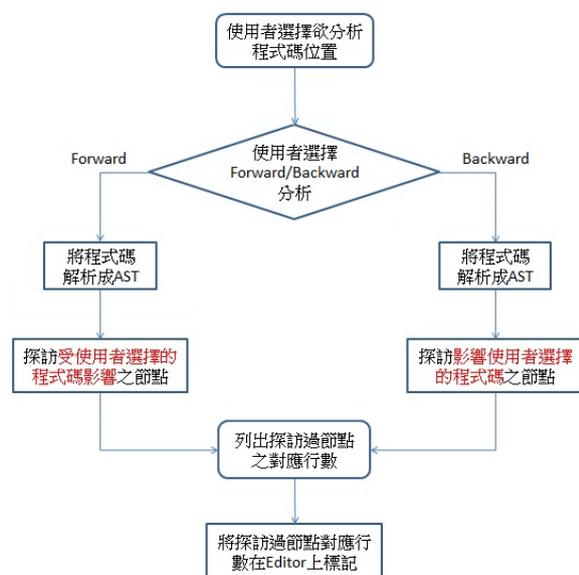
四、研究報告內容

(一) 研究範圍

架構於 Eclipse 上，使用 Java 語言開發 C++ Program Slicing Eclipse Plugin，用於對 C++ 程式碼做 Program Slicing，並追蹤程式碼。

(二) 流程架構及預期進行方式

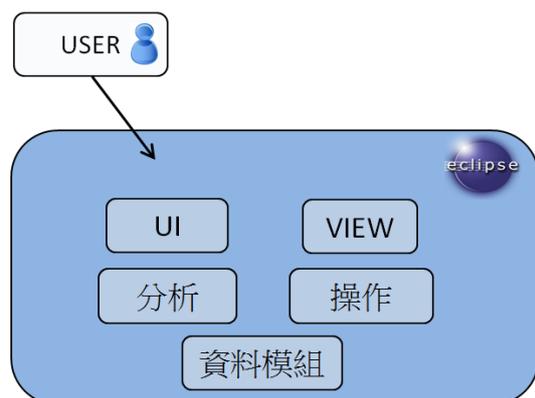
1. 操作流程：



圖一 操作流程圖

在 Eclipse 上開發 C++ 程式的時候，若需要使用 C++ Program Slicing Eclipse Plugin 輔助時，先選擇欲追蹤程式碼的位置，使用者可以點擊 Forward 或 Backward 的圖示選擇要進行 Forward 或 Backward 分析，此時程式會開始執行，最後將相關程式碼的位置在 Editor 上標記出來。

2. 系統架構：



圖二 系統架構圖

使用者點擊 Forward 或 Backward 圖示後，程式先呼叫分析的演算法，並對當前 editor 的程式碼分析，分析完成後針對分析結果進行操作和搜尋，將所需的資料擷

取出來，完成後返回到 UI，最後呼叫 VIEW 將所需要的結果顯示出來。

(三) 實驗結果

最後將專案包裝成一個 Eclipse Plugin，讓使用者在 Eclipse 上開發 C++ 程式，對程式進行測試、除錯時更佳的方便，並將使用者欲追蹤的相關程式碼標記在 Editor 上，例如：選了一個變數，再點擊 Forward 圖示，就能找到那些程式碼受選擇的變數影響，若點擊 Backward 圖示，找到那些程式碼影響到選擇的變數，最後在 Editor 上標記出來。如此一來，使用者便能快速地找到有 Bug 的程式碼，縮短使用者在龐大的程式碼裡找尋錯誤程式碼的時間。

五、參考文獻

- [1] Yossi Peery, Program Slicing for Refactoring. (<http://www.cs.tau.ac.il/~amiram/SoftwareSeminar/ProgramSlicingforRefactoring.ppt>)
- [2] JDK (<http://www.oracle.com/technetwork/java/javase/overview/index.html>)
- [3] Eclipse CDT (<http://www.eclipse.org/cdt/>)
- [4] MinGW (<http://www.mingw.org/>)
- [5] JDK、SDK、JRE、JVM 概念詳解 (<http://www.guan8.net/Java/12002.html>)
- [6] sungo, [教學] Window+GCC+CDT 用 Eclipse 開發 C、C++，2004. (<http://www.javaworld.com.tw/jute/post/view?bid=10&id=53262>)