

Coding Standard Tool

專題編號：113-CSIE-S003

執行期限：112 年第 1 學期至 113 年第 1 學期

指導教授：郭忠義

專題參與人員：110590028 黃冠鈞

110590002 王燦竑

110590034 楊榮鈞

110590048 陳俊諺

一、摘要

本專題使用 Python、Robot Framework、HTML、CSS 和 JavaScript，並結合 Flask 框架以及 Robocop，開發一個適用於 Sunbird DCIM - test automation 的靜態程式碼分析工具，此工具含有 Test Script Pseudocode IDE、Rules Checker，以網站呈現 IDE 的環境和 Rules Checker 的分析結果，讓使用者可以藉由 Test Script Pseudocode 快速產生 Robot Framework Test Script Draft 以及快速理解到所寫的程式碼是否有不符合 Coding Standard。

關鍵詞：靜態程式碼分析、Test Script Pseudocode、Coding Standard

二、緣由與目的

當團隊成員進入 Sunbird 時，經常遇到 Coding Standard 問題，導致單一測試完成時間延長，且回覆 PR Comment 時也因 Coding Standard 問題增加修復時間。減少檢查 Coding Standard 的時間，團隊決定開發輔助工具來提升測試效率及 PR 回覆速度。在撰寫 Robot Framework Script 前，團隊先將所需測試步驟撰寫為 Test Script Pseudocode 並交由 Product Owner 審核。然而，成員發現 Test Script Pseudocode 與實際編寫的腳本高度重複，於是決定將重複部分轉化為現有程式碼，以節省重複書寫的時間。

三、研究架構

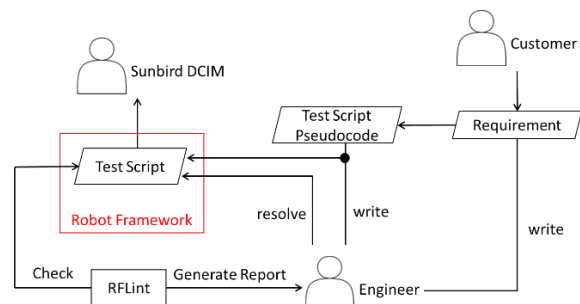


圖1. 原有的開發流程

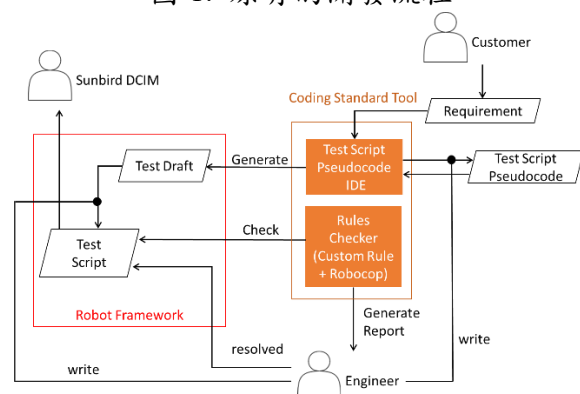


圖 2. 加入本專題的開發流程

原有開發流程中(圖 1)，將 Test Script Pseudocode 轉成 Test Script 和檢查 Coding Standard (RFLint 加上人工審查)的耗時較長，影響開發效率。加入本專題後(圖 2)，通過開發自動化工具，將減少轉成 Test Script 與檢查 Coding Standard 的時間。新的開發流程整合本專題開發之工具，優化開發流程，提升測試效率並縮短處理時間。

四、研究範圍

(一) Test Script Pseudocode 撰寫的標準化及自動化

1. 分析現有的 Test Script Pseudocode 撰寫方式，並制定標準最佳化撰寫規範。
2. 將 Test Script Pseudocode 轉換成

Robot Framework Test Script Draft，以加快測試的開發速度。

(二) 靜態程式碼分析技術

1. 將現有的 Coding Standard 規範整理成靜態程式碼分析的 Rule。
2. 根據 Rule 去分析 Robot Framework 腳本。

五、使用技術

(一) Test Script Pseudocode IDE

1. 採用正規表達式解析 Test Script Pseudocode 格式，並將解析的格式轉換成 Robot Framework Test Script Draft。
2. 與 Sunbird DCIM 資深開發者和其他團隊如 test automation 討論相關格式。

(二) Rules Checker

1. 本專題對 Reviewer 在 Pull Request 上的回覆，以及學長姐的經驗進行整理，將這些意見整合成 Coding Standard Rules。
2. 實作 Rules Checker 的方式，是依據 AST 架構掃描 Test Script 後送進 python 程式中檢查是否符合 Coding Standard Rules 的定義。
3. 結合 Robocop 現有的 Rule，並藉由其 API 輔助偵測整合的 Coding Standard Rules。

(三) AST 架構建立與分析

本專題使用 Robot Framework 的官方 API 解析出 Robot Framework 語言的 AST (Abstract Syntax Tree)，在取得 AST 之後，對裡面的語法、keyword 和變數名字等等做檢查，如圖 3。

六、實驗結果

藉由新加入的 Sunbird 團隊成員來使用此工具後，根據 Test Script Pseudocode IDE 的使用情況，撰寫 Test Script Pseudocode 的速度有顯著的提升，在轉成 Test Script Draft 的部分，也幫助撰寫 Test Script 的人有一個可以參考的模板，而 Rules Checker 的部分，讓新進成員更不容

易出現違反 Coding Standard 的情況。

七、結論

本專題做出 Test Script Pseudocode IDE，增加團隊開發的效率。同時藉由整合多個常有的 Coding Standard 和建議，成功做出各種檢查程式碼的 Rule，通過檢查這些 Rule，我們能夠減少修復 Coding Standard 的時間、提高程式碼審查效率。

未來預計會與新進的 Sunbird 團隊成員持續維護此工具，並根據 Sunbird 的工程師新提出的 Coding Standard 進行更新。

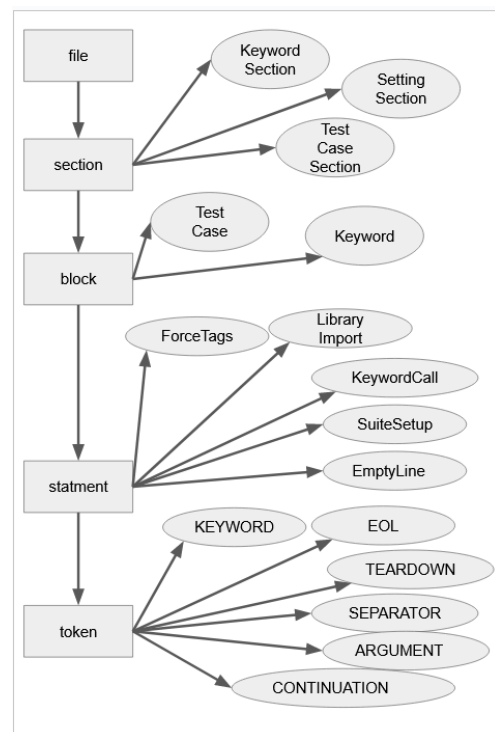


圖 3. AST 的各個類別

參考文獻

- [1] Robot Framework documentation
<https://robotframework.org/robotframework/>
- [2] Robot Framework API documentation
<https://robotframework.readthedocs.io/en/stable/>
- [3] Robocop
<https://robocop.readthedocs.io/en/stable/index.html>