國立臺北科技大學

資訊工程系碩士班

碩士論文

編號：107598051

改善”支援多國語言的Robot Framework

網頁自動化驗收測試工具”的翻譯邏輯、一詞多譯、代理關鍵字，並發佈使其完整

研究生：林稟宸



**資訊工程系碩士班**

**碩士學位論文**

**改善**

**”支援多國語言的Robot Framework網頁自動化驗收測試工具” 的翻譯邏輯、一詞多譯功能、代理關鍵字，並發佈使其完整**

**Improve Translation Logic, Multi-Translations Function, and Proxies of “Automated Web Testing Tool with Robot Framework which Support Internationalization“, and Release It**

**研究生：林稟宸**

**指導教授：鄭有進 教授、謝金雲 教授**

**中華民國一百一零年六月**

**摘要**

論文名稱：改善”支援多國語言的Robot Framework網頁自動化驗收測試工具”的翻譯邏輯、一詞多譯功能、代理關鍵字，並發佈使其完整

頁數：

校所別：國立台北科技大學 資訊工程系碩士班

畢業時間：一百零九學年度 第二學期

學位：碩士

研究生：林稟宸

指導教授：鄭有進 教授、謝金雲 教授

相關學術名詞：Robot Framework、xpath、關鍵字(Keyword)、代理關鍵字(Keyword Proxy)、i18n

以往基於Robot Framework的網頁驗收測試，同一份測試腳本的測試對象往往只限於一種語言的網頁，但現今的國際化社會，隨著商業模式的改變，一個網頁可能有多種語言的版本給不同國家的人使用。為了做出類似或相同的測試目的，測試者因此被迫撰寫更多重複性高的測試腳本，如此一來，測試的成本便會大大增加。

根據朱峻平的論文: ”支援多國語言的Robot Framework 網頁自動化驗收測試” (i18n)，透過使用者提供的JSON格式多國語言網頁翻譯檔，建立出一份單字對應翻譯的翻譯路徑檔，並在程式中定義翻譯的邏輯以及代理關鍵字，即可在不改動Robot Framework 測試腳本的情況下，完成多國語言網頁自動化驗收測試。

然而，現在的i18n工具仍然存在著許多待改善之處，例如:

1. 目前i18n工具只支援8種Robot Framework原生關鍵字，如果未來測試腳本使用到其他未支援的Robot Framework原生關鍵字，便會發生錯誤
2. 程式目前的翻譯對象僅限於網頁上的text(), normalize-space(),以及@title三種屬性，屆時，若使用者的測試腳本是用沒有列舉出來的屬性撰寫，例如:@placeholder, @arial-label等等，便會出錯。
3. 測試腳本運行期間，只要遭遇一詞多譯的情況，目前i18n工具僅於報表上列出該待翻譯詞的可能翻譯有哪些，並從中選取會使測試通過的翻譯詞當作當前翻譯。使用者無法自己選擇預期的翻譯詞去跑腳本測試。

此外，假如存在多個翻譯詞都會通過此測試(可能是因為xpath使用contains，使得畫面上要驗證的資訊只要包含於翻譯詞，測試就會通過；又或者畫面上剛好有多個元件符合翻譯後的測試腳本)，翻譯過後腳本的測試對象，就會偏離了使用者原先的預期。

1. 目前i18n工具尚未包裝成可以讓使用者直接安裝後使用的擴充工具，停留在使用者必須將github上的專案clone下來，並且在該專案上開發測試腳本的階段。如此會造成使用者的許多不便。

因此，此篇論文我將針對上述幾點待改善的地方，進行功能上的補強，希望藉此讓i18n工具在未來變得更易於使用。

**ABSTRACT**

Title：Improve Translation Logic, Multi-Translations Function, and Proxies of “Automated Web Testing Tool with Robot Framework which Support Internationalization“, and Release It

Pages：

School：National Taipei University of Technology

Department：Computer Science and Information Engineering

Time：June, 2021

Degree：Master

Researcher：Bing-Chen Lin

Advisor：Yu Chin Cheng Ph.D. and Chin-Yun Hsieh Ph.D.

Related terms：Robot Framework, xpath, Keyword, Keyword Proxy, I18n

In the past, the test object of one acceptance test script based on Robot Framework was often limited to web page in one language. However, nowadays a web page may have multiple language versions for people live in different countries. Therefore, tester often need to write more repetitive test scripts, and the cost of testing will also increase.

According to the thesis “Internationalization Support for Automated Web Testing with Robot Framework”(i18n) written by Chun-Ping Chu, i18n tool can build the translation mapping between words and their translations by JSON format translation files given by user. Besides, translation logic and proxies of keywords also let us can run the same web page acceptance test in different languages by the same test script without changing a word.

However, the current i18n tool still have many flaws to be improved, such as:

1. Currently, the i18n tool only support 8 Robot Framework native keywords. If other unsupported Robot Framework native keywords are used in the future, it may cause an error.
2. The current translation target of I18n tool are limited to text(), normalize-space(), and @title on the web page. So a test script will go wrong if it is written with attributes that are not listed above.
3. If a word have multiple translations currently, i18n tool only lists the possible translations on the test report, and selects the translation that would pass the test as the current solution. The user cannot choose the expected translation by himself to run the test script.

In addition, if there exist many translations would pass the test (maybe due to xpath is written by ‘contains()’, as long as the test target is included in the xpath, the test will pass. Or there are multiple components on the web page that match the translation), the way we take currently may deviate from the original expectation of the user.

1. At present, installation is not supported by the i18n tool, so users can only clone the project from github and develope their own test scripts of Robot Framework inside the project. This is not user friendly.

Therefore, I am going to improve the defects of the I18n tool that I mentioned above, hoping that the i18n tool will be easier to use in the future.

**誌 謝**

**目 錄**

[摘要 i](#_Toc42761836)

[ABSTRACT iii](#_Toc42761837)

[誌 謝 v](#_Toc42761838)

[目 錄 vi](#_Toc42761839)

[表目錄 ix](#_Toc42761840)

[圖目錄 x](#_Toc42761841)

[第一章 緒論 1](#_Toc42761842)

[1.1 研究背景與動機 1](#_Toc42761843)

[1.2 研究目標 1](#_Toc42761844)

[1.3 論文組織架構 2](#_Toc42761845)

[第二章 背景知識 3](#_Toc42761846)

[2.1 國際化 3](#_Toc42761847)

[2.2 網頁前端框架分析 3](#_Toc42761848)

[2.3 自動化驗收測試 5](#_Toc42761849)

[2.4 Robot Framework 5](#_Toc42761850)

[2.4.1 Robot Framework腳本 6](#_Toc42761851)

[2.4.2 Robot Framework產生之報表 8](#_Toc42761852)

[2.4.3 Robot Framework Listener 9](#_Toc42761853)

[第三章 研究方法與實作 10](#_Toc42761854)

[3.1 系統設計 10](#_Toc42761855)

[3.1.1 原生類別 11](#_Toc42761856)

[3.1.2 本論文方法所設計的類別 11](#_Toc42761857)

[3.1.3 系統執行流程 13](#_Toc42761858)

[3.2 建立翻譯查找路徑對照表 14](#_Toc42761859)

[3.3 根據翻譯查找路徑對照表包裝翻譯邏輯 18](#_Toc42761860)

[3.3.1 翻譯一個單字 18](#_Toc42761861)

[3.3.2 翻譯多個單字 20](#_Toc42761862)

[3.3.3 翻譯xpath內的單字 20](#_Toc42761863)

[3.4 產生各關鍵字的代理 21](#_Toc42761864)

[3.4.1 分析關鍵字的代理 24](#_Toc42761865)

[3.5 產生報表提示 27](#_Toc42761866)

[3.5.1 顯示xpath翻譯過後的資訊 27](#_Toc42761867)

[3.5.2 顯示單字翻譯過後的資訊 28](#_Toc42761868)

[3.5.3 處理一詞多譯問題 28](#_Toc42761869)

[3.5.3.1 使用warning提示xpath內單字具有一詞多譯問題 29](#_Toc42761870)

[3.5.3.2 使用warning提示參數具有一詞多譯問題 31](#_Toc42761871)

[3.5.3.3 消除一詞多譯warning提示 32](#_Toc42761872)

[第四章 測試案例分析 35](#_Toc42761873)

[4.1 關鍵字使用到的xpath含有待翻譯單字 35](#_Toc42761874)

[4.2 關鍵字使用到待翻譯單字 38](#_Toc42761875)

[4.3 具有一詞多譯問題 39](#_Toc42761876)

[4.3.1 使用warning提示xpath內單字具有一詞多譯問題 40](#_Toc42761877)

[4.3.2 使用warning提示參數具有一詞多譯問題 42](#_Toc42761878)

[4.3.3 消除一詞多譯warning提示 44](#_Toc42761879)

[4.4 測試案例結果與討論 45](#_Toc42761880)

[第五章 結論與未來展望 46](#_Toc42761881)

[5.1 本論文方法的限制 46](#_Toc42761882)

[5.2 結論 46](#_Toc42761883)

[5.3 未來展望 47](#_Toc42761884)

[參考文獻 48](#_Toc42761885)

**表目錄**

[表格 2‑1網頁前端框架分析 3](#_Toc42761886)

[表格 3‑1原生類別 11](#_Toc42761887)

[表格 3‑2本論文方法所設計的類別 11](#_Toc42761888)

[表格 3‑3目前已提供代理的關鍵字 24](#_Toc42761889)

[表格 3‑4尚未提供代理的關鍵字 25](#_Toc42761890)

**圖目錄**

[圖2‑1 angular-translate-英文翻譯檔 4](#_Toc42761891)

[圖2‑2 angular-translate-繁體中文翻譯檔 4](#_Toc42761892)

[圖2‑3 HTML內加入標記 4](#_Toc42761893)

[圖2‑4 Robot Framework腳本 7](#_Toc42761894)

[圖2‑5擴充測試函式庫(self\_util.py) 7](#_Toc42761895)

[圖2‑6 Robot Framework產生之報表 8](#_Toc42761896)

[圖2‑7 Robot Framework Listener version 9](#_Toc42761897)

[圖3‑1系統類別圖 10](#_Toc42761898)

[圖3‑2 Robot Framework Listener參數設置圖 13](#_Toc42761899)

[圖3‑3英文翻譯檔範例 15](#_Toc42761900)

[圖3‑4繁體中文翻譯檔範例 15](#_Toc42761901)

[圖3‑5翻譯查找路徑對照表 16](#_Toc42761902)

[圖3‑6建立翻譯查找路徑對照表循序圖 17](#_Toc42761903)

[圖3‑7 value method 19](#_Toc42761904)

[圖3‑8 get\_possible\_translation method 19](#_Toc42761905)

[圖3‑9 values method 20](#_Toc42761906)

[圖3‑10 translate method 20](#_Toc42761907)

[圖3‑11代理關鍵字Should Be Equal的循序圖 22](#_Toc42761908)

[圖3‑12代理關鍵字的擴充介面 23](#_Toc42761909)

[圖3‑13 xpath翻譯過後的資訊 26](#_Toc42761910)

[圖3‑14單字翻譯過後的資訊 27](#_Toc42761911)

[圖3‑15報表上方欄提示xpath內單字具有一詞多譯問題 28](#_Toc42761912)

[圖3‑16詳細提示xpath內單字具有一詞多譯問題 28](#_Toc42761913)

[圖3‑17報表上方欄提示參數具有一詞多譯問題 30](#_Toc42761914)

[圖3‑18詳細提示參數具有一詞多譯問題 30](#_Toc42761915)

[圖3‑19消除一個單字的一詞多譯warning提示之參數設定 31](#_Toc42761916)

[圖3‑20消除多個單字的一詞多譯warning提示之參數設定 31](#_Toc42761917)

[圖3‑21使用檔案消除多個單字的一詞多譯warning提示之參數設定 32](#_Toc42761918)

[圖3‑22消除xpath內單字一詞多譯warning提示 32](#_Toc42761919)

[圖3‑23消除參數一詞多譯warning提示 33](#_Toc42761920)

[圖4‑1 xpath含有待翻譯單字More的自動化測試腳本 34](#_Toc42761921)

[圖4‑2單字More顯示於英文頁面 35](#_Toc42761922)

[圖4‑3單字More顯示於繁體中文頁面 35](#_Toc42761923)

[圖4‑4關鍵字(Keyword) Get Text After It Is Visible 36](#_Toc42761924)

[圖4‑5 xpath含有待翻譯單字More的報表資訊呈現 36](#_Toc42761925)

[圖4‑6關鍵字使用到待翻譯單字More的自動化測試驗證腳本 37](#_Toc42761926)

[圖4‑7關鍵字使用到待翻譯單字More報表資訊呈現 37](#_Toc42761927)

[圖4‑8具有一詞多譯問題-英文頁面 38](#_Toc42761928)

[圖4‑9具有一詞多譯問題-繁體中文頁面 39](#_Toc42761929)

[圖4‑10 xpath內單字Support具有一詞多譯問題的自動化測試腳本 39](#_Toc42761930)

[圖4‑11報表上方欄提示xpath內單字Support具有一詞多譯問題 40](#_Toc42761931)

[圖4‑12詳細提示xpath內單字Support具有一詞多譯問題 40](#_Toc42761932)

[圖4‑13關鍵字(Keyword)參數Support具有一詞多譯問題的自動化測試腳本 41](#_Toc42761933)

[圖4‑14報表上方欄提示參數Support具有一詞多譯問題 42](#_Toc42761934)

[圖4‑15詳細提示參數Support具有一詞多譯問題 42](#_Toc42761935)

[圖4‑16消除單字Support的一詞多譯warning提示之參數設定 43](#_Toc42761936)

[圖4‑17消除xpath內單字Support一詞多譯warning提示 43](#_Toc42761937)

[圖4‑18消除參數Support一詞多譯warning提示 43](#_Toc42761938)

**第一章 緒論**

* 1. **研究背景與動機**

根據朱峻平的論文: ”支援多國語言的Robot Framework 網頁自動化驗收測試” (i18n)，透過使用者提供的JSON格式多國語言網頁翻譯檔，建立出一份單字對應翻譯的翻譯路徑檔，並在程式中定義翻譯的邏輯以及代理關鍵字，即可在不改動Robot Framework 測試腳本的情況下，完成多國語言網頁自動化驗收測試。

然而，現在的i18n工具仍然存在著許多待改善之處，會造成使用上的困難，以下分別列出四點: 支援的Robot Framework原生關鍵字只有八種、翻譯邏輯只支援三種html屬性、一詞多譯無法讓使用者自己選擇、該工具尚未發佈並支援安裝。

* 1. **研究目標**

本論文旨在改善i18n工具上述提及的四項缺陷，希望透過我的努力，可以讓此I18n工具在未來執行多國語言網頁自動化驗收測試時，更易於使用。

而實作的目標分別列舉如下:

1. Robot framework版本剩下的代理關鍵字，使得所有常用原生關鍵字都能正確的翻譯
2. 透過”負面表列法”，將xpath中確定不會執行翻譯的屬性(例如:@id , @class等等)儲存於list中，若腳本執行到當下的xpath需要被翻譯，且其中有屬性不在list中，則利用這些屬性們生成一個新的翻譯規則，再執行翻譯。
3. 執行完含有一詞多譯的測試腳本後，如果測試通過，便開啟一個圖形化介面，記錄了執行翻譯時當下關鍵字的參數組合，並顯示所有可能的翻譯詞，讓使用者可以從中去選擇，並產生一個設定檔。之後再次執行測試腳本時，i18n工具便會根據設定檔的內容去選擇適當的翻譯詞，同時消除報表上的warning提示
4. 將I18n工具包裝成為可以pip install 的library
   1. **論文組織架構**

本論文分為五個章節，第一章介紹研究背景與動機，以及期望達成的研究目標。第二章介紹本論文相關的背景知識。第三章探討本論文的研究方法與實作，詳述擴充代理關鍵字的流程、改善翻譯邏輯的完整思路，以及如何藉由圖形化使用者介面去改善i18n工具目前遇到一詞多譯時的處理方法。第四章展示實際的測試案例，記錄做了上述改動後，使用者執行網頁自動化驗收測試時，和先前的i18n工具相比，有甚麼差異之處。第五章則是結論以及未來展望，概述本論文的成果以及尚且存在的使用限制和待改善之處。

1. **第二章 背景知識**
   1. **國際化**

國際化，英文讀作Internationalization，簡稱i18n，18意謂著i和n之間的18個英文字母。國際化是開發軟體時，將軟體本身和特定語言、地區脫鉤的一個過程，除了可以滿足不同的地區、文化的大眾需求，移植軟體時也不須改變內部程式的實作。

* 1. **自動化驗收測試**

驗收測試，英文讀作Acceptance Testing，是一種站在使用者的立場，去檢驗一個真實存在的系統，是否滿足使用者需求與預期的測試方法。

而透過撰寫自動化測試腳本，如今我們可以執行更符合成本，且更加精確的自動化驗收測試。改善了過往手動驗收測試存在的人為操作誤差、成本較高、系統問題無法即時呈現等問題。

* 1. **Robot Framework**

Robot Framework 是一個開源的框架語言，可以用來執行自動化驗收測試或機器人自動化，核心框架是由Python編寫而成，測試者可以使用Python或Java擴充其函式庫。其特色是擁有簡單的語法，以及容易理解的原生關鍵字(Keyword)，測試者可以視需求使用並包裝成更接近自然語言的關鍵字。

* + 1. **Robot Framework測試腳本**

一個Robot Framework 的測試腳本基本上由三個區塊構成，分別是

1. Settings: 包含了使用到的library與resource file，也可以將Test Setup(test case執行前要做的動作)、Test Teardown(test執行後要做的動作)定義於此。
2. Test Cases: 在此測試者可以為各項想要驗證的使用者需求，撰寫核心的測試腳本
3. Keywords: 如果測試腳本即將使用並非Robot Framework的原生關鍵字，或未被定義於其他resource file中，測試者可以在此處撰寫出新的關鍵字去達到測試目的。
   * 1. **Robot Framework測試報表**

在測試腳本運行結束後，Robot Framework會產生出一份測試報表，記錄了整體的執行狀況(通過或失敗)以及時間，透過可展開的階層式圖表，可以方便測試者去追蹤到當前出現問題的關鍵字，進而修復錯誤。

* 1. **XPath**

XPath，全名為 XML Path Language，可以用來定位XML檔案中某節點所處在的位置。在用Robot Framework撰寫的網頁自動化驗收測試中，我們便時常需要藉助XPath，去找到並確定畫面上某元件的位置，以對其狀態進行測試。

1. **第三章 研究方法與實作**

本章節將詳述如何針對i18n工具現存的四項問題，進行思考與改善，並與朱峻平的論文”支援多國語言的Robot Framework網頁自動化驗收測試” (第一版i18n工具)進行比較，呈現出先後的差異。

* 1. **系統設計**

圖3‑1為第一版i18n工具的系統類別圖，經過部分功能改善，並為其擴充代理關鍵字後，使其成為圖3-2當前版本的i18n系統類別圖。

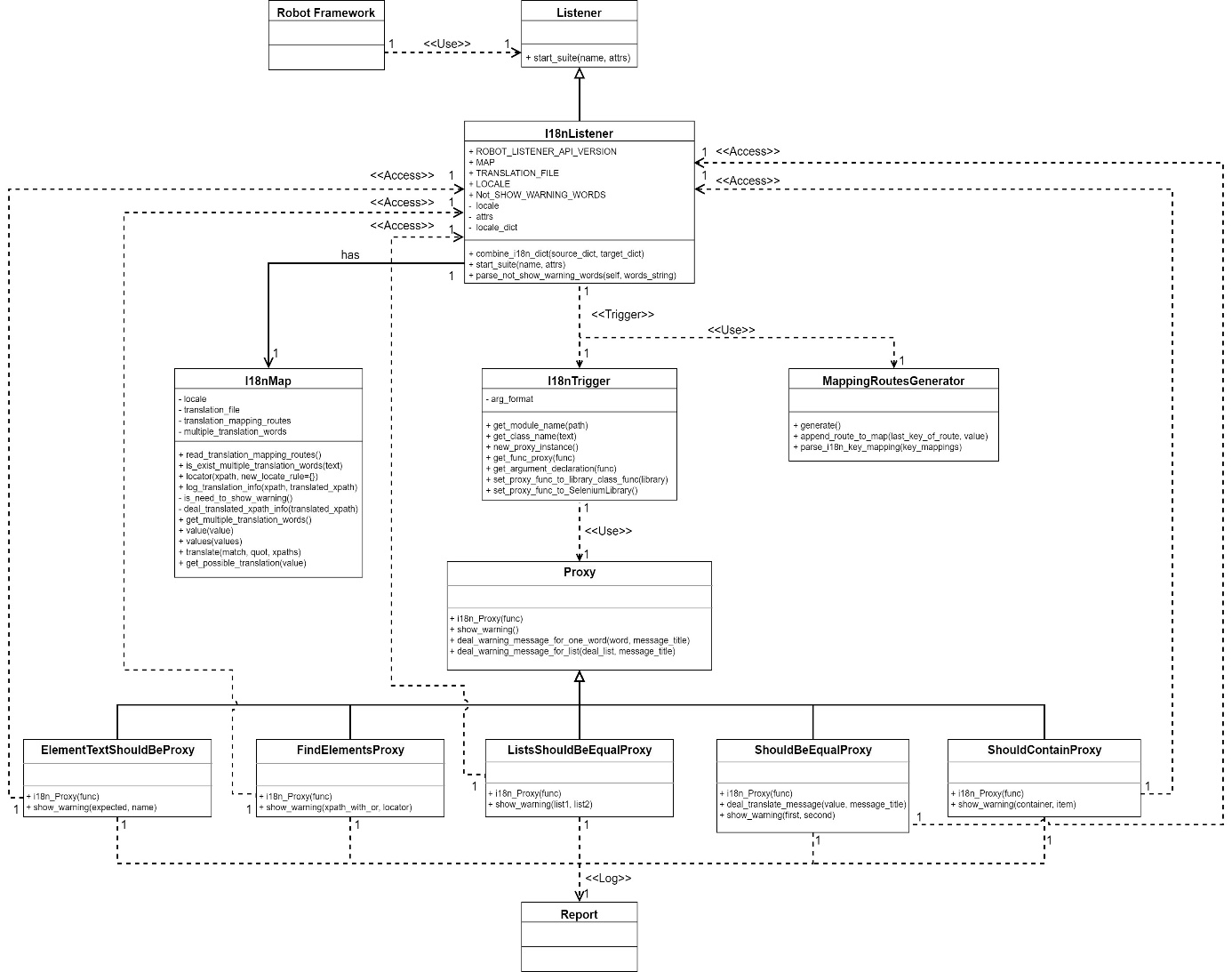


圖3‑1系統類別圖

以下將以概述的方式，介紹Robot Framework的原生類別，與i18n第一版已存在的類別(詳細可參考朱峻平的論文3-1)。並以表格3-1的方式呈現本論文所新增加的類別。

* + 1. **第一版i18n工具的現存類別與原生類別**

Robot Framework的原生類別分別有RobotFramework，Listener，和Report。

第一版i18n設計的類別則有I18nListener，MappingRoutesGenerator，I18nMap，I18nTrigger，Proxy，以及五個繼承Proxy的代理關鍵字類別: ElementTextShouldBeProxy，FindElementProxy，ListsShouldBeEqualProxy，ShouldBeEqualProxy，ShouldContainProxy。

* + 1. **本論文新擴充的類別**

表格 3-1 本論文新擴充的類別

|  |  |
| --- | --- |
| **類別名稱** | **類別功能** |
| UI | 在程式執行期間，記錄下遭遇到一詞多譯的詞語以及其翻譯。在程式執行後，產生一詞多譯UI介面供使用者選擇。 |
| AlertShouldBePresentProxy | 繼承自父類別Proxy並改寫其實作，負責代理”Alert Should Be Present”關鍵字的參數翻譯邏輯，並將翻譯好的參數部分回傳給原關鍵字。 |
| CountValuesInListProxy | 繼承自父類別Proxy並改寫其實作，負責代理”Count Values In List”關鍵字的參數翻譯邏輯，並將翻譯好的參數部分回傳給原關鍵字。 |
| DictionariesShouldBeEqualProxy | 繼承自父類別Proxy並改寫其實作，負責代理”Dictionaries Should Be Equal”和”Dictionary Should Contain Sub Dictionary”關鍵字的參數翻譯邏輯，並將翻譯好的參數部分回傳給原關鍵字。 |
| DictionaryShouldContainItemProxy | 繼承自父類別Proxy並改寫其實作，負責代理”Dictionary Should Contain Item”關鍵字的參數翻譯邏輯，並將翻譯好的參數部分回傳給原關鍵字。 |
| DictionaryShouldContainKeyProxy | 繼承自父類別Proxy並改寫其實作，負責代理”Dictionary Should Contain Key”關鍵字的參數翻譯邏輯，並將翻譯好的參數部分回傳給原關鍵字。”Dictionary Should Contain Item”其內部實作也會使用到此Proxy。 |
| DictionaryShouldContainValueProxy | 繼承自父類別Proxy並改寫其實作，負責代理”Dictionary Should Contain Value”關鍵字的參數翻譯邏輯，並將翻譯好的參數部分回傳給原關鍵字。 |
| GetMatchCountProxy | 繼承自父類別Proxy並改寫其實作，負責代理”Get Match Count”關鍵字的參數翻譯邏輯，並將翻譯好的參數部分回傳給原關鍵字。 |
| ListSelectionShouldBeProxy | 繼承自父類別Proxy並改寫其實作，負責代理”List Selection Should Be”關鍵字的參數翻譯邏輯，並將翻譯好的參數部分回傳給原關鍵字。 |
| ListShouldContainSubListProxy | 繼承自父類別Proxy並改寫其實作，負責代理”List Should Contain Sub List”關鍵字的參數翻譯邏輯，並將翻譯好的參數部分回傳給原關鍵字。 |
| ListShouldContainValueProxy | 繼承自父類別Proxy並改寫其實作，負責代理”List Should Contain Value”和”List Should Not Contain Value”關鍵字的參數翻譯邏輯，並將翻譯好的參數部分回傳給原關鍵字。 |
| ListShouldNotContainDuplicatesProxy | 繼承自父類別Proxy並改寫其實作，負責代理”List Should Not Contain Duplicates”關鍵字的參數翻譯邏輯，並將翻譯好的參數部分回傳給原關鍵字。 |
| RemoveFromDictionaryProxy | 繼承自父類別Proxy並改寫其實作，負責代理”Remove From Dictionary”關鍵字的參數翻譯邏輯，並將翻譯好的參數部分回傳給原關鍵字。 |
| RemoveValuesFromListProxy | 繼承自父類別Proxy並改寫其實作，負責代理”Remove Values From List”關鍵字的參數翻譯邏輯，並將翻譯好的參數部分回傳給原關鍵字。 |
| SelectFromListByLabelProxy | 繼承自父類別Proxy並改寫其實作，負責代理”Select From List By Label”, “Unselect From List By Label”關鍵字的參數翻譯邏輯，並將翻譯好的參數部分回傳給原關鍵字。 |
| SelectFromListByValueProxy | 繼承自父類別Proxy並改寫其實作，負責代理”Select From List By Value”, “Unselect From List By Value”關鍵字的參數翻譯邏輯，並將翻譯好的參數部分回傳給原關鍵字。 |
| TableCellShouldContainProxy | 繼承自父類別Proxy並改寫其實作，負責代理”Table Cell Should Contain”關鍵字的參數翻譯邏輯，並將翻譯好的參數部分回傳給原關鍵字。 |
| TableColumnShouldContainProxy | 繼承自父類別Proxy並改寫其實作，負責代理”Table Column Should Contain”關鍵字的參數翻譯邏輯，並將翻譯好的參數部分回傳給原關鍵字。 |
| TableRowShouldContainProxy | 繼承自父類別Proxy並改寫其實作，負責代理”Table Row Should Contain”關鍵字的參數翻譯邏輯，並將翻譯好的參數部分回傳給原關鍵字。 |
| TableShouldContainProxy | 繼承自父類別Proxy並改寫其實作，負責代理”Table Should Contain”,”Table Header Should Contain”,“Table Footer Should Contain”關鍵字的參數翻譯邏輯，並將翻譯好的參數部分回傳給原關鍵字。 |
| TitleShouldBeProxy | 繼承自父類別Proxy並改寫其實作，負責代理“Title Should Be”關鍵字的參數翻譯邏輯，並將翻譯好的參數部分回傳給原關鍵字。 |

* + 1. **系統執行流程與比較**

第一版的i18n其執行流程，首先使用者必須於Red編輯器中的Additional Robot Framework arguments設定參數，例如:-d out –L debug -–listener code/listeners/I18nListener.py:zh-TW。

測試腳本執行前，系統會去呼叫I18nListener.py，當前語言zh-TW代表繁體中文-台灣。之後讀取由使用者自行提供的JSON格式翻譯檔，建立出一份”Key階層->翻譯”的翻譯路徑檔，以及建立出所有所有自定義的代理關鍵字物件。測試腳本執行時，每當運行到一個有定義代理之關鍵字，就會去呼叫該特定的代理關鍵字物件，並執行翻譯邏輯。測試腳本結束後，將一詞多譯的warning資訊顯示在報表上，若想隱藏特定的翻譯詞資訊，可以事後在手動在組態設定。

新版的i18n執行流程，基本沿用了第一版的架構，但擴充了代理關鍵字(詳見3-2)，也優化了翻譯邏輯(詳見3-3)。

而改變較大的部份，是在測試腳本結束後，如果有遭遇過一詞多譯，便會跳出一詞多譯的圖形化使用者介面(詳見3-4)，記錄了執行翻譯當下關鍵字的參數組合，並顯示所有可能的翻譯詞，讓使用者可以從中去選擇並產生一個設定檔。之後再次執行測試腳本時，系統便會根據設定檔的內容去選擇適當的翻譯詞，同時消除報表上的warning提示資訊。

* 1. **擴充代理關鍵字使其完整支援原生關鍵字**

在此前，第一版i18n工具只支援8種Robot Framework原生關鍵字(如圖3-2)。尚有32種需要翻譯的原生關鍵字，未提供支援(如圖3-3)。如果測試腳本使用到這些未支援的原生關鍵字去跑多國語言網頁測試，則會導致出錯。因此，本論文的解法是擴充完目前Robot Framework版本剩下的代理關鍵字，使得所有原生關鍵字的參數部分都能正確得被翻譯。

且又因為先前的Robot Framework版本是3.0.4，SeleniumLibrary版本是3.1.1，而現在的Robot Framework版本已更新到3.2.2，SeleniumLibrary版本更新到4.5.0，因此，原先需要被支援的32種原生關鍵字，部分已被淘汰剩下了，只剩下30種。

以下將以XXX Proxy為例，詳述在現今配合一詞多譯的架構下，我是如何完成擴充代理關鍵字的，以及和第一版i18n代理關鍵字的實作差異。

圖

* 1. **改善翻譯邏輯使其能應對各種html屬性**

在先前的翻譯邏輯中，若一個XPath內部存在多種html屬性，i18n工具使用了

列舉法(如圖3-4)，僅提供@title、text()、normalize-space()三種屬性可以成功的被執行翻譯。屆時，若測試腳本的XPath是用沒有列舉出來的屬性撰寫，但卻有翻譯的需求，例如:@placeholder、@arial-label等等，則會導致測試出錯。

為了解決以上問題，本論文早期設想了大致三種解決方向:

1.

2.

3.

在多方優缺點的考量下，本論文最後採用了第三種”負面表列法”去解決問題，因為此種作法相較於第一種解法顧及到了系統執行效能，來的更加省時。相較於第二種類似於AI的作法，在實作的複雜度上，則更為簡單，且把是否翻譯的決定權交給了使用者，而非全部靠系統。相較於原本的”把翻譯規則用列舉法固定”相比，則更能顧及到未來測試腳本的不確定性，因為我們無法預期使用者會用怎麼樣的方式去編寫XPath。

以下將詳述新版i18n工具改善後的翻譯邏輯，

* 1. **提供圖形化使用者介面解決一詞多譯的問題**

先前遇到一詞多譯問題時，i18n工具提供了warning資訊於測試結束後的報表上，此作法確實提醒了使用者該腳本存在著一詞多譯問題，且也有顯示目前系統採用的翻譯詞，這是好的部分，本論文將繼續沿用。此翻譯邏輯是站在一個”最大化讓測試通過”的立場，假如遇到一詞多譯，系統會將所有的翻譯詞都執行看看。直到一包可能的翻譯詞中，出現了一個會使測試通過的翻譯詞，則算測試通過，並將該翻譯詞呈現於報表上。

然而，此作法也存在著一個缺陷，就是”使用者無法自由的選擇真正想要的翻譯詞”去執行腳本測試，而把決定權全權授予系統。並會有以下兩點問題:

1.

2.

所以，本論文將提供一個圖形化使用者介面，記錄下執行翻譯時當下的關鍵字參數組合，並顯示所有可能的翻譯詞，讓使用者從中去選擇，並產生一個設定檔，以便之後再次執行同一測試腳本時，根據設定檔的內容去選擇適當的翻譯詞。期望透過如此的方法來改善上述兩點一詞多譯所遭遇的難題。

以下，將詳述一詞多譯UI的整體實作與使用時機:

* 1. **將i18n工具設計成可以安裝的library**

本論文透過xxx工具，將i18n包裝成為一個可以直接讓使用者安裝的Python library。解決了先前使用者只能從github上將i18n工具clone下來，並在該專案上開發測試腳本的不方便。

以下將詳述實際的步驟:

1. **第四章 測試案例分析**

本章節將為第三章擴充的代理關鍵字撰寫單元測試，並且將改善翻譯邏輯與一詞多譯功能後的i18n工具，做功能的展示，最後套用至一個使用多項代理關鍵字的Robot Framework測試腳本，以呈現 ”多國語言網頁自動化驗收測試”的效果。

* 1. **新增之代理關鍵字的單元測試**

以下將為各個新增之代理關鍵字，分別提供Robot Framework的測試腳本，以驗證關鍵字在新的擴充代理下，能夠獨立正常運作。但為求精簡，本論文將不重複放上同類型關鍵字的測試腳本。

圖4‑1 Alert Should Be Present在代理關鍵字下執行的測試腳本與測試結果

圖4‑2 xxx在代理關鍵字下執行的測試腳本與測試結果

圖4‑3 xxx在代理關鍵字下執行的測試腳本與測試結果

圖4‑4 xxx在代理關鍵字下執行的測試腳本與測試結果

圖4‑5 xxx在代理關鍵字下執行的測試腳本與測試結果

圖4‑6 xxx在代理關鍵字下執行的測試腳本與測試結果

圖4‑7 xxx在代理關鍵字下執行的測試腳本與測試結果

圖4‑8 xxx在代理關鍵字下執行的測試腳本與測試結果

圖4‑9 xxx在代理關鍵字下執行的測試腳本與測試結果

* 1. **改善翻譯邏輯後的驗收測試示例**

以下將舉xxx 關鍵字的測試腳本為例，說明在改善XPath翻譯邏輯前後，執行含有@arial-label屬性的XPath之腳本，將分別得到何種結果

圖xx 待翻譯XPath運行在未改善翻譯邏輯時的i18n工具

圖xx 待翻譯XPath運行在改善翻譯邏輯後的i18n工具

由以上的結果我們可以發現，…

* 1. **一詞多譯情況下所產生的圖形化使用者介面**

圖4‑8為遭遇一詞多譯情況時，系統於腳本結束時所產生的圖形化使用者介面，上面分別記錄了遭遇一詞多譯的關鍵字完整參數，以及所有可能的翻譯詞，供使用者做選擇。當時用者選擇希望了翻譯詞並按下Submit按鈕後，系統便會產生一份設定檔，以利下次執行到同一份腳本的同一關鍵字時，直接套用選擇的翻譯，並消除掉一詞多譯的問題。

以下將分別說明一詞多譯的圖形化使用者介面，在不同情況下，會有甚麼不同的反應，與是否會出現:

圖xx 為在測試腳本不通過的情況下，不管是否遭遇一詞多譯

圖xx 為在測試腳本通過且沒有一詞多譯的情況下，或是設定檔中已經定義了使用者對特定詞語的翻譯詞

圖xx 為在測試腳本通過且遭遇一詞多譯的情況下

* 1. **使用多項代理關鍵字的測試腳本**

以上介紹的測試案例，分別呈現了本論文各項改動的實際成果，但因為都偏向單一關鍵字或單一功能的測試，在此將設計一個較大型的Robot Framework網頁自動化驗收測試腳本，內部使用到多項代理關鍵字，來對XX的中文版網頁進行一系列操作，同時，會遭遇到一詞多譯，以及使用相別於i18n第一版所列舉的html待翻譯屬性，來撰寫XPath。藉此來對新版i18n工具作一個驗收。

以下將執行兩次測試腳本，分別呈現選擇一詞多譯翻譯詞前，與選擇翻譯詞後，

最後測試腳本的結果。

圖XX 為對XX網頁的測試腳本

圖XX 網頁測試腳本第一次執行結果

圖XX 網頁測試腳本第二次執行結果

1. **第五章 結論與未來展望**
   1. **本論文限制**

i18n工具尚存在著一定的使用限制，其原因部份源自於特定的原生關鍵字其性質屬用對畫面上進行取值，如果該關鍵字一旦遭遇一詞多譯的情況，便會導致測試腳本的錯誤，所以解法只能從代理關鍵字內部下手…

* 1. **結論**

本論文提出了四項措施去改進第一版i18n工具所面臨的問題，分別是XXXX。透過以上努力，即可讓今後的多國語言網頁自動化驗收測試腳本，不再侷限於只能使用特定的關鍵字去撰寫。且XPath的撰寫也不再需要考慮只有@title，text()，normalize-space()會被翻譯的問題，使用者的測試腳本可以用更多元的XPath寫法來搭配此i18n工具使用。遭遇一詞多譯時，使用者也能夠在第一次測試腳本結束後，選擇內心所預期的翻譯詞來執行接下來的迴歸測試。最後，藉由將i18n工具包裝成可以pip install 的Python library，使用者可以隨安裝即享受到本工具所提供的功能，而無須將自己的測試腳本移植到i18n工具的資料夾下。

* 1. **未來展望**

目前i18n工具的代理關鍵字尚存在著使用的限制，未來可以朝XXX的方向去改善其實作。且一詞多譯UI部分，當使用者選擇到錯誤的翻譯詞時，沒有提供一個可以回覆原始設定(Undo)的機制，造成使用上的不便，未來可以朝向豐富一詞多譯UI介面的功能去努力；記錄翻譯詞的部分，目前也只有記錄關鍵字的參數，少數情況下可能會和其他的關鍵字發生重疊，未來可以將關鍵字名稱一併記錄下來，以提高處理一詞多譯時的精確性。進而讓「多國語言網頁自動化驗收測試」工具變得更貼近測試者的需要，錯誤率低且容易使用。**參考文獻**

1. 李允中，軟體工程，臺北：社團法人臺灣軟體工程學會，2013，第194頁
2. Paul M. Duvall, Steve Matyas, Andrew Glover, Continuous Integration,

New Jersey: [Pearson Education (US)](https://www.bookdepository.com/publishers/Pearson-Education-US), 2007.

1. 鍾承翰，基於 Scrum 與 Mob programming 開發自動化驗收測試的準樣式，

碩士論文，國立臺北科技大學資訊工程系碩士班，臺北，2020。

1. 呂昭陞，一個改善XPath表達式在網頁應用程式自動化驗收測試之穩定性的方

法，碩士論文，國立臺北科技大學資訊工程系碩士班，臺北，2018。

1. [Chin-Yun Hsieh](https://ieeexplore.ieee.org/author/38258708400), [Chen-Hsin Tsai](https://ieeexplore.ieee.org/author/37073881500), [Yu Chin Cheng](https://ieeexplore.ieee.org/author/38226238700). (2013, May). Test-Duo: A framework

for generating and executing automated acceptance tests from use cases, San Francisco, CA, USA

1. “INTERNATIONALIZATION” [Online]. Available:

<https://www.w3.org/standards/webdesign/i18n>.

1. “Localization vs. Internationalization” [Online]. Available:

<https://www.w3.org/International/questions/qa-i18n>.

1. “JSON - Introduction” [Online]. Available:

<https://www.w3schools.com/js/js_json_intro.asp>.

1. “Robot Framework” [Online]. Available: [https://robotframework.org/.](https://robotframework.org/)
2. “Robot Framework User Guide” [Online]. Available:

<https://robotframework.org/robotframework/latest/RobotFrameworkUserGuide>.

1. “The Python Tutorial” [Online]. Available:

<https://docs.python.org/release/3.6.0/tutorial/index.html.>

1. “Microsoft official website” [Online]. Available: <https://www.microsoft.com/>.
2. “angular-translate” [Online]. Available:

<https://www.microsoft.com/>.

1. “ng-i18next” [Online]. Available:

https://github.com/i18next/ng-i18next.

1. “angular-i18n” [Online]. Available:

https://github.com/nolazybits/angular-i18n.

1. “ngx-translate” [Online]. Available:

https://github.com/ngx-translate/core.

1. “angular-i18next” [Online]. Available:

https://github.com/Romanchuk/angular-i18next.

1. “Transloco” [Online]. Available:

https://github.com/ngneat/transloco.

1. “react-i18next” [Online]. Available:

https://github.com/i18next/react-i18next.

1. “redux-i18n” [Online]. Available:

<https://github.com/APSL/redux-i18n>.