

SideeX: 擴充 Selenium IDE 測試案例錄製與撥放自動化能力

SideeX: Expanding Selenium IDE Record-Playback Automation Capability

李信杰^{1,2}、黃琪恩²、游傑麟²、陳與賢²、石碩亨²

¹ 國立成功大學計算機與網路中心

² 國立成功大學資訊工程學系

Shin-Jie Lee, Chi-En Huang, Chieh-Lin Yu, Yu-Xian Chen, Shuo-Heng Shih
Email: jleee@mail.ncku.edu.tw, {tony4794, ygl0118, chenuxian, baimao8437}@gmail.com

摘要

Selenium IDE 是一個熱門的網頁測試案例自動化錄製與播放集成開發環境，然而此工具目前仍存在著許多錄製遺漏與錯誤待人工進行修正之限制。本研究主要目標為發展 Selenium IDE 一系列擴充套件(稱作 SideeX—Selenium IDE Extended)，強化 Selenium IDE 自動錄製與播放(Test Case Record-Playback)能力，並包裝成為 Mozilla Firefox 的附加元件供網頁測試者免費下載與使用。此擴充版本除了包含核心元素定位擴充(Web Element Locator Extension)之外，亦包括未命名視窗與內嵌框架元素定位擴充、指令執行之明確等待擴充、AJAX 等待擴充、具改變網頁元素之鼠標移動錄製擴充、具改變網頁元素之捲動錄製擴充、滑鼠拖曳擴充。實驗結果顯示 SideeX 之核心元素定位擴充具有高正確率，搭配其他擴充能大幅降低使用者手動修正測試案例頻率。

關鍵字: Selenium IDE、網頁元素定位(Web Element Locating)、測試案例錄製、網頁功能性測試

1. 前言

在 Selenium[1]中，測試案例由一連串的命令組成，而這些命令的目標元素則由下列定位器產生：Xpath、DOM、CSS、LinkText、ID、Name。然而，在利用 Selenium IDE 記錄下來的網頁應用中，459 個 ID 定位器有小於 2%無法使用，655 個 Name 定位器、473 個 LinkText 定位器、357 個 CSS 定位器大約有 12%至 20%無法使用，而 791 個 Xpath 定位器則大約有 60%無法使用[2]。除此之外，在平常使用的經驗上，從數百個網站收集的 4000 個 Xpath 定位器中，絕對路徑和相對路徑分別僅有 47%和 76%可以穩定運作[3]。近年來之研究方向其中一個根本問題為，給定一個舊版本網頁、舊版本網頁的元素、更改內容後的新版本網頁，如何準確地在新版本網頁定位該元素。本研究主要目標為發展 Selenium IDE 一系列擴充套件(稱作 SideeX—Selenium IDE Extended)，強化 Selenium IDE 自動錄製與播放能力，並包裝成為 Mozilla Firefox 的附加元件供網頁測試者免費下載與使用。

2. SideeX 功能介紹

SideeX 包含下列擴充：

- CoreElementLocatorExt：定位持續演變的網頁中的元素(畫面參照圖 1)。原始的 Selenium IDE 定位元素是利用 ID、CSS、Xpath 等屬性，而一個持續演變的網頁屬性值可能有所改變，若指令依據單一的屬性值定位而該屬性變動時，指令就會出錯；我們發展了一個新穎分析技術，並能有效提高定位元素的正確率。
- UnnamedWindowAndIFrameExt：定位未命名的視窗或內嵌框架中的元素。原始的 Selenium IDE 若視窗或內嵌框架並未命名，錄製和播放時就會出錯；此擴充自動給予未命名視窗或內嵌框架一個序列編號或特定名稱，使錄製和播放順利執行。
- ExplicitWaitExt：在執行每一個指令前強制等待一段特定時間。原始的 Selenium IDE 在執行指令時不會等待網頁讀取，而會直接執行下一個指令，若下一個指令需要未讀取完成的網頁資料時，執行指令就會出錯；此擴充強制所有指令執行前等待一段特定時間，以讓網頁讀取完畢。
- AJAXWaitExt：強制等待 AJAX 請求執行完畢。原始的 Selenium IDE 在執行 AJAX 後不會等待傳送完成，而會直接執行下一個指令，若下一個指令需要從伺服器回傳的資料時，執行指令就會出錯；此擴充強制指令觸發 AJAX 後，等待 AJAX 請求執行完成。
- InfluentialMouseoverExt：定位鼠標移動。網頁中某些元素會因為鼠標移動而新增，若指令需要鼠標移動後新增的元素時，原始的 Selenium IDE 執行指令就會出錯。此擴充基於 DomNodeInserted 的事件驅動記錄鼠標移動。
- InfluentialScrollingExt：定位在捲動至底部時會動態讀取的網頁中的元素。網頁中某些元素會因為捲動至底部而新增，原始的 Selenium IDE 無法錄製捲動，若指令需要捲動後新增的元素時，執行指令就會出錯。此擴充將會模擬捲動的行為。
- DragAndDropExt：Selenium IDE 可以執行 dragAndDrop 的動作，但是卻沒有辦法錄製，擴充後會根據使用者 Mousedown 的元素及拖曳後 Mouseup 的位置，將操作紀錄成 dragAndDrop 的指令。

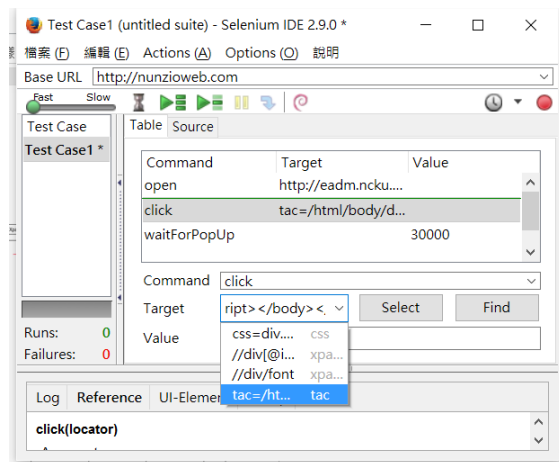


圖 1 SideeX 核心元素定位擴充畫面(藍色部分)

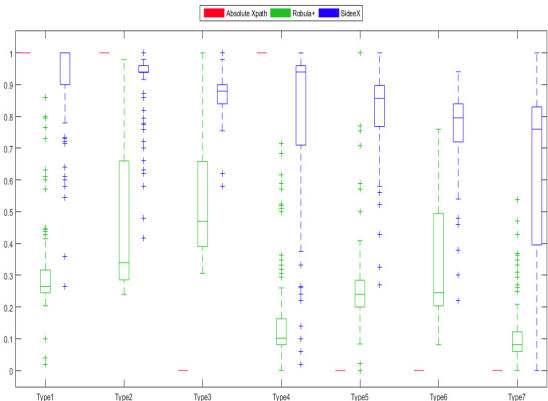


圖 2 Absolute XPath(紅)、Robula+(綠)與 SideeX(藍)網頁元素定位(Web Element Locating)正確率比較表

表 1 Selenium IDE、Robula+與 SideeX 於三個改版網站之測試案例錄製後播放之準確率比較

	Selenium IDE												Robula+	SideeX	
	id	link	name	css	dom: name	xpath: link	xpath: attributes	xpath: idRelative	xpath: href	dom: index	xpath: position	The First Locator			
Bugzilla 4.0 => Bugzilla 5.0	21/21	20/26	12/12	31/33	12/12	20/26	22/23	22/48	21/25	0/15	33/48	41/48	45/48	48/48	
Moodle 1.9 => Moodle 2.6	5/6	10/11	5/5	15/20	0/0	12/16	5/9	3/29	0/16	0/0	10/29	24/29	23/29	28/29	
Wordpress 3.0 => Wordpress 4.0	27/29	9/9	25/26	29/36	11/19	7/13	29/31	28/45	9/14	0/20	32/45	38/45	30/45	45/45	
Total	53/56	39/46	42/43	75/89	23/31	39/55	56/63	53/122	30/55	0/35	75/122	103/122	98/122	121/122	
Accuracy													84.4%	80.3%	99.2%

3. 實驗評估

SideeX 最重要的核心價值為發展出一個新穎的核心元素定位擴充技術(Web Element Locator Extension)。我們針對此擴充設計了兩部分實驗驗證：(1)模擬驗證：更改舊版本網頁中隨機元素的文字、屬性、位置後存成新網頁，驗證該元素是否能準確地定位；(2)真實案例驗證：經由現行不同版本的網頁，驗證舊版本的元素是否能準確地定位。

在模擬驗證中，我們於 79 個真實網頁上進行七種更動模擬：更改文字 (Type 1)，更改屬性 (Type 2)，更改位置 (Type 3)，更改文字、屬性 (Type 4)，更改文字、位置 (Type 5)，更改屬性、位置 (Type 6)，更改文字、屬性、位置 (Type 7)。並比較 Absolute XPath、Robula+、SideeX 三種定位器定位能力，實驗結果參照圖 2。Absolute XPath 定位結果很極端，位置未改變時，定位正確率高達 100%；而位置改變時，定位則完全沒有正確率。SideeX 在元素內容更改的情況下，正確率仍能超越 Robula+ 之表現。

在真實案例驗證中，我們利用 Selenium IDE、Robula+、SideeX 三種定位器錄製 Bugzilla、Moodle、Wordpress 三個網站舊版之測試案例，並運行這些測試案例於此三個網站新版上。測試結果參照表 1，Selenium IDE 和 Robula+ 的準確率分別為 84.4% 和 80.3%，而 SideeX 的準確率為 99.2%。

4. 結論

一個新版本網頁可能因設計者改變，或是因使用者互動而改變，不管是何種原因導致元素定位失敗，其根本問題都是如何定位元素，本研究的目的即是解決這類重要且常見的問題，目前研究成果 SideeX 包含核心元素定位擴充、未命名視窗與內嵌框架元素定位擴充、指令執行之明確等待擴充、AJAX 等待擴充、具改變網頁元素之鼠標移動錄製擴充、具改變網頁元素之捲動錄製擴充、滑鼠拖曳擴充。實驗結果顯示 SideeX 之核心元素定位擴充具有高正確率，能大幅降低使用者手動修正測試案例頻率，提升 Selenium IDE 自動化測試案例錄製與播放之實用性。

致 謝

感謝科技部編號 MOST 103-2221-E-006-218 之計畫對本研究之經費提供與技術支援，由於科技部的支持，使本研究得以順利進行，特此致上感謝之意。

參考文獻

- [1] Selenium. <http://www.seleniumhq.org/>.
- [2] M. Leotta, D. Clerissi, F. Ricca, and P. Tonella, "Capture-Replay vs. Programmable Web Testing: An Empirical Assessment during Test Case Evolution," *Proceedings of the 20th IEEE Working Conference on Reverse Engineering*, pp. 272-281, 2013.
- [3] M. Kowalkiewicz, M. E. Orlowska, T. Kaczmarek, and W. Abramowicz, "Robust Web Content Extraction," *Proceedings of the 15th International Conference on World Wide Web*, 2006.