Repairing Selenium Test Cases: An Industrial Case Study about Web Page Element Localization

Maurizio Leotta, Diego Clerissi, Filippo Ricca, Cristiano Spadaro

Abstract:

This poster presents an industrial case study about test automation and test suite maintenance in the context of Web applications. The Web application under test is a Learning Content Management System (eXact learning LCMS). We analysed the costs associated with the realignment of four equivalent Selenium WebDriver test suites, implemented using the page object pattern and different methods to locate web page elements, to a subsequent release of eXact learning LCMS. In our study, the two ID-based test suites required significantly less maintenance effort than the XPath-based ones.

Digital Object Identifier (DOI):

`http://dx.doi.org/10.1109/ICST.2013.73`

Repairing Selenium Test Cases:
An Industrial Case Study about Web Page Element Localization

Maurizio Leotta[1], Diego Clerissi[1], Filippo Ricca[1], Cristiano Spadaro[2]
[1] *Dipartimento di Informatica, Bioingegneria, Robotica e Ingegneria dei Sistemi (DIBRIS), Università di Genova, Italy*
[2] *eXact learning solutions S.p.A., Sestri Levante, Italy*
maurizio.leotta@unige.it, diego.clerissi@gmail.com, filippo.ricca@unige.it, c.spadaro@exactls.com

***Abstract*—This poster presents an industrial case study about test automation and test suite maintenance in the context of Web applications. The Web application under test is a Learning Content Management System (eXact learning LCMS). We analysed the costs associated with the realignment of four equivalent Selenium WebDriver test suites, implemented using the page object pattern and different methods to locate web page elements, to a subsequent release of eXact learning LCMS. In our study, the two ID-based test suites required significantly less maintenance effort than the XPath-based ones.**

***Keywords*-Web Application Testing, Test Automation, Selenium WebDriver, Test Suite Maintainability.**

I. Introduction

Test suite maintenance tends to have the biggest impact on the overall cost for test automation [1]. This is particularly true for Web applications (WAs) because: (1) they are subject to continuous changes and rapid evolution that often "break" one or more test cases and (2) realigning (repairing) a test suite is a time-consuming and expensive task.

The framework used in this case study for automating WA testing is **Selenium WebDriver**[1]. It is part of Selenium, a suite employed in many industrial projects [2], [3]. Selenium WebDriver provides a comprehensive programming interface used to control a browser as a real user does, and the test cases are implemented manually in a programming language integrating Selenium WebDriver commands with JUnit or TestNG assertions. Selenium WebDriver offers several **Methods for Locating Web page Elements** (e.g., links, input fields). The ID-based method locates the web page elements using their id values. The XPath-based method selects a web page element by navigating the HTML tree. Finally, the LinkText-based method allows to select a link in a web page searching for its text (since it works only for links, it has to be used in conjunction with the other methods). The ***goal*** of this work is understanding which method for locating web page elements (e.g., "search by ID", "search by XPath") reduces the maintenance effort needed to realign the test cases to a new release.

Often, Selenium WebDriver test suites are implemented using the **Page Object pattern**[2]. The page object pattern is used to model the web pages involved in the test process as objects, employing the same programming language used to write the test cases. In this way, the functionalities offered by a web page become "services" (i.e., methods) offered by the corresponding page object and can be easily called within any test case. Thus, all the details of the web page are encapsulated inside the page object. This allows the test developer to work on a higher level of abstraction and helps to reduce the coupling between test cases and web pages.

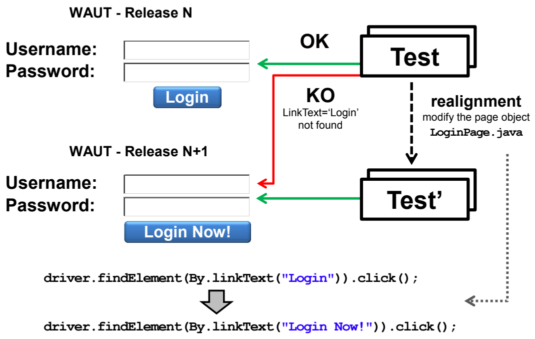
II. The Case Study

The **Web Application Under Test** (WAUT) is eXact learning LCMS. It is mainly a Learning Content Management System for eLearning content production developed in ASP.NET. The development started about 6 years ago, with a team composed by 3-4 software analysts and developers. eXact learning LCMS is composed by about 700.000 lines of code and 200 ASP.NET pages.

Compared Test Suites: the test team focused on a portion of eXact learning LCMS to test with Selenium WebDriver. They chose the DURP portion, managing Domains, Users, Roles and Permissions that can be defined in the application. eXact learning LCMS has not been developed with the idea of automating the test cases (e.g., no meaningful ids are available). The developers of eXact learning LCMS used a facility of Visual Studio IDE that adds an auto-generated id for each HTML tag: these values are strings concatenated with numbers generated automatically (e.g., id1, id2, id3, ... , idN). Making use of these auto-generated values, the "search by ID" test suite was developed for the M9 release of eXact learning LCMS. Then, starting from this test suite, we built three equivalent test suites using the XPath and LinkText methods obtaining the following test suites: "search by ID+LinkText", "search by XPath" and "search by XPath+LinkText". These new test suites are equivalent to the "search by ID" one; they test the same functionalities in the same way. Only the page objects are different, since different methods to locate the web page elements are used. XPath expressions have been automatically built using specific tools. Each test suite is composed of 25 test cases and 19 page objects for a total of 3320 Java LOCs (1720 for the test cases and 1600 for the page objects). Overall, in the 19

[1] http://seleniumhq.org/projects/webdriver/

[2] http://seleniumhq.org/docs/06_test_design_considerations.jsp



```
driver.findElement(By.linkText("Login")).click();
driver.findElement(By.linkText("Login Now!")).click();
```

Figure 1. "Login" link changes in "Login Now!" ⇒ Realignment required

page objects, we have 131 web page element localization lines (i.e., the point in the code where a page object calls the driver to find a web page element). Each test case of our test suite for eXact learning LCMS is composed by several actions such as: navigating a web page, clicking a link, filling a form and finally, evaluating a set of assertions. They were implemented using conditional statements, loops, logging functionality and exception handling; all of them are parameterized (a.k.a. data-driven) test cases. When the test suite is executed, 336 test case instances are run (each test cases is parameterized) in about 3,5 hours.

Research Question: we compared the four test suites trying to answer to the following RQ: *Which is the more effective localization method among "search by ID", "search by XPath", "search by ID+LinkText" and "search by XPath+LinkText" for reducing the effort to realign a test suite when a new version of the WAUT is created?*. In other words, we are interested to determine whether one (or more) of the proposed methods for locating web page elements is clearly better than the other ones. We measured the realignment effort in terms of time (minutes) and number of LOCs to change for all the four considered test suites.

III. Summary of the Results

As a first step, we ran the four test suites against the new release of eXact learning LCMS (M10), observing that all the 25 test cases composing the four test suites failed. Then, we realigned each test case composing the test suites (see the example in Fig. 1), noting down the time required for repairing it and recording the number of LOCs modified. During the realignment, we discovered that the changes made in eXact learning LCMS (M10) did not affect the logic of the test cases. For this reason, we modified only the page objects to realign the four test suites (more precisely, we modified only a subset of the 131 localization lines).

Fig. 2 shows, by means of a scatter-plot, the time required to realign the test cases belonging to the four test suites to the new release of eXact learning LCMS[3]. It is evident that the time spent for realigning XPath-based test cases goes beyond the time for realigning the ID-based ones in almost all the cases. Moreover, it is interesting to note that the trend of the time required for realigning the test cases is similar for all the four test suites (high for the first test cases and low for the subsequent ones, see the regression curves in Fig. 2). The first test cases required more time to be realigned than the subsequent ones since the page object pattern has been adopted. Indeed, to realign the first test cases of each test suite (e.g., test cases 1, 2, 3 and 4), we modified the page objects they use. But, these modifications were also useful for the subsequent test cases (e.g., 5, 6, and 7), thus they required less time to be realigned.

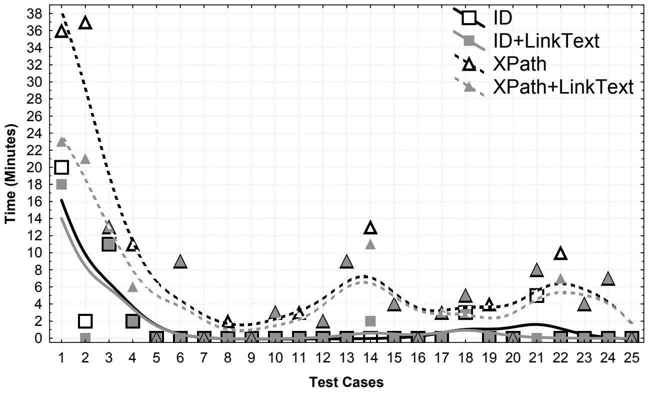


Figure 2. Time of test cases realignment

The answer to our research question can be easily deduced by Fig. 3 that shows some comparative data about the four implemented test suites: ID-based test suites have required the lowest maintenance effort, while the XPath-based test suites have required a higher effort (about 4 times more for what concerns the time and 10 times more for the number of LOCs to modify). In both cases the combined use of ID and XPath with LinkText has slightly improved the results.

	Time		Code	
Test Suites	Minutes	Difference	LOCs	Difference
ID	43		9	
ID+LinkText	36	- 16 %	8	- 11 %
XPath	183	+ 326 %	96	+ 967 %
XPath+LinkText	135	+ 214 %	86	+ 856 %

Figure 3. Effort required using different Web Page localization methods. The difference is computed considering ID as baseline

IV. Conclusion

The main result of our case study is that, from the point of view of realignment effort, ID-based methods for locating web page elements are better than XPath methods. These results are not conclusive, but this work compares for the first time in a real industrial context, four Selenium test suites built using different methods for locating web page elements.

References

[1] S. Berner, R. Weber, and R. Keller. Observations and lessons learned from automated testing. In *Proc. of ICSE 2005*, pages 571–579. IEEE, 2005.

[2] B. Haugset and G. Hanssen. Automated acceptance testing: A literature review and an industrial case study. In *Proc. of AGILE 2008*, pages 27–38. IEEE, 2008.

[3] M. Leotta, D. Clerissi, F. Ricca, and C. Spadaro. Improving test suites maintainability with the page object pattern: An industrial case study. In *Proc. of ICSTW 2013*. IEEE, 2013.

[3] The four regression curves have been computed using the distance-weighted least squares method.