Web-Test-Tool-Evaluation

Inhalt

Synopsis

This paper documents the evaluation of different tools to test an AJAX-application based on ExtJS. The evaluation started because of strong problems with the currently used test-tool named “Sahi”. Candidates as replacement for Sahi are:

- Windmill Testing Framework
- Selenium web application testing system
- QF-Test

The main focus of this evaluation is capture and replay as this is the common way how new testers will start writing tests. Refactoring of tests for better maintainability is discussed in short side notes.

Purpose

The purpose of this evaluation is to find one common test-tool for all tests of web applications at CoreMedia. This way the knowledge of the usage of this test-tool should be sharable between different projects.

Requirements

The main requirements for the new test-tool as derived from GUI Testing – Tool Requirements and Test Guidelines:

1. ***Short time to learn***
   New testers should soon be able to understand and debug existing tests. Writing first basic tests should be simple.
2. ***Multiple browser-support***
   Browsers which should (at least) be supported:
   - Internet Explorer 7 and 8
   - Firefox 3
3. ***Easy debugging***
   This includes that automatic test-runs provide screenshots in case of errors so that the possible error can easily be detected. And it includes that tests can be debugged step by step and that they can be run independently. The latter one of course also poses a requirement to the tests which are written – not only to the tool.
4. ***Clipboard Interaction***
   An important part during our tests is interaction with the clipboard. Thus the tools should be able to record clipboard interaction events.

Optional and untested during this evaluation is a comparison of screenshots (expected versus actual). Although important for the tests as layout problems are a major threat with multi-browser-support the tests are complicate to develop to keep them robust not only on multiple browsers but also on multiple platforms. It is known that QF-Test offers such a screenshot-comparison which can be easily recorded.

Test Fixture

The application under test (AUT) is a modified version of the application provided in the ExtJS Blog. Modifications are:

- ***Each component got an explicit ID set.***
  This is because tests are more robust with IDs set. Otherwise tests will rely on the DOM-structure (for example) which might easily be refactored in later releases. I prefixed these IDs with "id-" just to detect more easily how and where the test-tools use and store the IDs.
- ***Date and Time field got duplicated.***
  This is because both open a "popup" to select a value. The first test is what ExtJS does with these popups: Will they be reused for other components of the same type? The answer is no. And the second test is that the test-tool should be able to handle those two popups of the same time.

The new introduced Date field got named “Update” and the new Time field got named “Other Time”. That’s important to know when reading the following test case.

Test Case

1. Load the ExtJS form.
2. Click into the Name Field
3. As Name enter: “John Doe”
4. Type Tab (to: Email-Field)
5. Check that the Email-Field is focused

   This is to test e. g. tab-order in the application.
6. Click right before the "D" of "John Doe" in the Name field
7. Enter “Jane “ (with trailing space)

   Typically web-applications listen to keystroke events e. g. to count characters typed. Thus the test-tool should be able to replay keystroke events.
8. Type Tab (again to: Email-Field)
9. Check that text of Name field is “John Jane Doe”
10. Enter “john-jane.doe@anywhere.com“
11. Click into Company Field
12. As Company enter “Replace Me”
13. Open Company-ComboBox by click on Arrow Down

    Interesting in this evaluation: Is the test-tool able to identify this button to click in a robust way.
14. Select “Google Inc.” by clicking on it.
15. Check that the text entered is “Google Inc.”
16. Click into Birthday field.
17. As Birthday enter 01/01/2000.
18. Click on Calendar Button
19. Check that Calendar opens
20. Check that Saturday January 1st, 2000 is opened.
21. Select January 31st, 2000 by clicking on it.
22. Check that Calendar closes
23. Check that field value is 01/31/2000
24. Click into “Update”-Field
25. Enter Date 01/01/2000
26. Click on Calendar Button
27. Click on Today-Button
28. Check that Calendar closes
29. Check that current date is entered.
30. Click on Arrow Down of Time-Field.
31. Click on time 1:00 AM
32. Check that time 1:00 AM is entered.
33. Click into "Other Time" Field
34. Enter “1:00”
35. Check that type-ahead-box shows 1:00 AM and 1:00 PM
36. Hit Cursor Down and Enter to select “1:00 PM”

37. Check Checkboxes Classical and Blues
38. Select Favorite Color “Blue”
39. Fill in Description:

    Tests the clipboard interaction.

    1. Triple Click Name field
    2. Check that complete Name is selected
    3. Press Ctrl+C
    4. Click into Description Field
    5. Paste with Ctrl+V
    6. Check that Text is the same as in Name field
40. Done.

Test Recording

Symbols

- ✓ Test passed successfully.
- ✗ Test failed.
- ~ Test partially passed; e. g. recorded actions do not match performed actions.

Selenium

1. *Load the ExtJS form.* – ✓ Piece of cake.
2. *Click into the Name Field* – ✗ Click is not recorded.
3. *As Name enter: “John Doe”* – ~ Name is not entered via Keystrokes but value is set.
4. *Type Tab (to: Email-Field)* – ✗ Tab not recorded.
5. *Check that the Email-Field is focused* – ✗ Focus check not possible. Requires changes to the web application to listen to onfocus-events.
6. *Click right before the "D" of "John Doe" in the Name field* – ✗ Click not recorded.
7. *Enter “Jane “ (with trailing space)* – ✗ Keystrokes not recorded. Instead “Set value to John Jane Doe” is recorded.
8. *Type Tab (again to: Email-Field)* – ✗ Tab not recorded.
9. *Check that text of Name field is “John Jane Doe”* – ~ Check recording only possible with right click into web application. This sets the focus back to the name field which is bad and it poses a problem if the web application overrides the right click for showing a custom context menu for example.
10. *Enter “john-jane.doe@anywhere.com“* – ✓
11. *Click into Company Field* – ✗ Click is not recorded.
12. *As Company enter “Replace Me”* – ✓
13. *Open Company-ComboBox by click on Arrow Down* – ✗ ExtJS only provides a generic ID (“ext-gen61” for example) for this button. As Selenium records this ID it fails to replay this action with a new generic ID. Of course this can be changed later by another identification mechanism (XPath for example).
14. *Select “Google Inc.” by clicking on it.* – ✓
15. *Check that the text entered is “Google Inc.”* – ✓ (despite the focus constraint mentioned above)
16. *Click into Birthday field.* – ✗ Click is not recorded.

17. *As Birthday enter 01/01/2000.* – ✓
18. *Click on Calendar Button* – ✘ again problem with generic ID during replay
19. *Check that Calendar opens* – ~ It is not possible to record this. As workaround you may record that elements of the calendar appeared.
20. *Check that Saturday January 1st, 2000 is opened.* – ✘ It is not possible from within the IDE to record applied styles and classes which is required to detect for example the selected day.
21. *Select January 31st, 2000 by clicking on it.* – ✓
22. *Check that Calendar closes* – ✘ not possible from within the IDE
23. *Check that field value is 01/31/2000* – ✓
24. *Click into "Update"-Field* – ✘ Click is not recorded.
25. *Enter Date 01/01/2000* – ✓
26. *Click on Calendar Button* – ✘ again problem with generic ID during replay
27. *Click on Today-Button* – ✘ generic ID problem
28. *Check that Calendar closes* – ✘ not possible from within the IDE
29. *Check that current date is entered.* – ✓ Just date checked needs to be replaced by a dynamic date later during refactoring.
30. *Click on Arrow Down of Time-Field.* – ✘ generic ID problem
31. *Click on time 1:00 AM* – ~ Time has no ID… thus it is only recorded as selecting the 5th div.
32. *Check that time 1:00 AM is entered.* – ✓
33. *Click into "Other Time" Field* – ✘ Click is not recorded.
34. *Enter "1:00"* – ✓
35. *Check that type-ahead-box shows 1:00 AM and 1:00 PM* – ✘ Problem with required right click to perform check.
36. *Hit Cursor Down and Enter to select "1:00 PM"* – ✘ Neither cursor down, enter nor selecting/entering the specified time is recorded.
37. *Check Checkboxes Classical and Blues* – ✘ generic ID problem
38. *Select Favorite Color "Blue"* – ✘ generic ID problem
39. Fill in Description:
    1. *Triple Click Name field* – ✘ Triple click is not recorded.
    2. *Check that complete Name is selected* – ✘ Selection check not possible within IDE.
    3. *Press Ctrl+C* – ✘ not recorded
    4. *Click into Description Field* – ✘ not recorded
    5. *Paste with Ctrl+V* – ✘ not recorded; it is only recorded that the field value got set to "John Jane Doe"
    6. *Check that Text is the same as in Name field* – ✓
40. Done.

Windmill

1. *Load the ExtJS form.* – ~ file-URL not possible. Had to setup a simple webserver such as HFS to access the form via http.
2. *Click into the Name Field* – ✓
3. *As Name enter: "John Doe"* – ~ Name is not entered via Keystrokes but value is set.
4. *Type Tab (to: Email-Field)* – ✘ Tab not recorded.
5. *Check that the Email-Field is focused* – ✘ Checks from within the IDE do not seem to be possible. Need to be added manually later on.
6. *Click right before the "D" of "John Doe" in the Name field* – ✘ Click not recorded.

7. *Enter “Jane “ (with trailing space)* – ✘ Keystrokes not recorded. Instead “Set value to John Jane Doe” is recorded.
8. *Type Tab (again to: Email-Field)* – ✘ Tab not recorded.
9. *Check that text of Name field is “John Jane Doe”* – ✘ no checks while recording
10. *Enter “john-jane.doe@anywhere.com“* – ✓
11. *Click into Company Field* – ✓
12. *As Company enter “Replace Me”* – ✓
13. *Open Company-ComboBox by click on Arrow Down* – ✘ Just as Selenium Windmill has a problem with the generic IDs and thus replay will fail.
14. *Select “Google Inc.” by clicking on it.* – ✓
15. *Check that the text entered is “Google Inc.”* – ✘ no checks while recording
16. *Click into Birthday field.* – ✓
17. *As Birthday enter 01/01/2000.* – ✓
18. *Click on Calendar Button* – ✘ again problem with generic ID during replay
19. *Check that Calendar opens* – ✘ no checks while recording; in addition during my tests the calendar layout was broken (headline had no background; reproducible on that very day) – although I was not able to reproduce this some days later it gives a bad impression if the test tool modifies the web application in a way that it is corrupted.
20. *Check that Saturday January 1st, 2000 is opened.* – ✘ no checks while recording
21. *Select January 31st, 2000 by clicking on it.* – ✘ Click not recorded. Windmill does not seem to be able to identify the popup.
22. *Check that Calendar closes* – ✘ no checks while recording
23. *Check that field value is 01/31/2000* – ✘ no checks while recording
24. *Click into “Update”-Field* – ✓
25. *Enter Date 01/01/2000* – ✓
26. *Click on Calendar Button* – ✘ generic ID problem
27. *Click on Today-Button* – ✘ click not recorded again
28. *Check that Calendar closes* – ✘ no checks while recording
29. *Check that current date is entered.* – ✘ no checks while recording
30. *Click on Arrow Down of Time-Field.* – ✘ generic ID problem
31. *Click on time 1:00 AM*– ✓
32. *Check that time 1:00 AM is entered.* – ✘ no checks while recording
33. *Click into "Other Time" Field* – ✓
34. *Enter “1:00”* – ✓
35. *Check that type-ahead-box shows 1:00 AM and 1:00 PM* – ✘ no checks while recording
36. *Hit Cursor Down and Enter to select “1:00 PM”* – ✘ Neither cursor down nor enter is recorded. Instead Windmill records that the field value got set to “1:00 PM”.
37. *Check Checkboxes Classical and Blues* – ✘ not recorded
38. *Select Favorite Color “Blue”* – ✘ not recorded
39. Fill in Description:
    1. *Triple Click Name field* – ✓
    2. *Check that complete Name is selected* – ✘ no checks while recording
    3. *Press Ctrl+C* – ✘ not recorded
    4. *Click into Description Field* – ✓
    5. *Paste with Ctrl+V* – ✘ not recorded; it is only recorded that the field value got set to “John Jane Doe”

   6. *Check that Text is the same as in Name field* – ✗ no checks while recording

40. Done.

QF-Test

1. *Load the ExtJS form.* – ✓ Piece of cake with the Quick-Start-Wizard.
2. *Click into the Name Field* – ✓
3. *As Name enter: "John Doe"* – ✓ Single events are send (can be switched to "set value" with one click)
4. *Type Tab (to: Email-Field)* – ✓
5. *Check that the Email-Field is focused* – ~ Focus check not possible while recording. But it can easily be added after recording as QF-Test has its own proxy for DOM elements which offers to query if the element is focused.
6. *Click right before the "D" of "John Doe" in the Name field* – ✓
7. *Enter "Jane " (with trailing space)* – ✓
8. *Type Tab (again to: Email-Field)* – ✓
9. *Check that text of Name field is "John Jane Doe"* – ✓
10. *Enter "john-jane.doe@anywhere.com"* – ✓
11. *Click into Company Field* – ✓
12. *As Company enter "Replace Me"* – ✓
13. *Open Company-ComboBox by click on Arrow Down* – ✓
14. *Select "Google Inc." by clicking on it.* – ✓
15. *Check that the text entered is "Google Inc."* – ✓
16. *Click into Birthday field.* – ✓
17. *As Birthday enter 01/01/2000.* – ✓
18. *Click on Calendar Button* – ✓
19. *Check that Calendar opens* – ~ Hard to access the calendar DIV although QF-Test highlights the HTML-regions currently hovered above when recording checks.
20. *Check that Saturday January 1st, 2000 is opened.* – ✗ It is not possible from within the IDE to record applied styles and classes which is required to detect for example the selected day.
21. *Select January 31st, 2000 by clicking on it.* – ✓
22. *Check that Calendar closes* – ✗ not possible while recording
23. *Check that field value is 01/31/2000* – ✓
24. *Click into "Update"-Field* – ✓
25. *Enter Date 01/01/2000* – ✓
26. *Click on Calendar Button* – ✓
27. *Click on Today-Button* – ✓
28. *Check that Calendar closes* – ✗ not possible while recording
29. *Check that current date is entered.* – ✓
30. *Click on Arrow Down of Time-Field.* – ✓
31. *Click on time 1:00 AM* – ✓
32. *Check that time 1:00 AM is entered.* – ✓
33. *Click into "Other Time" Field* – ✓
34. *Enter "1:00"* – ✓
35. *Check that type-ahead-box shows 1:00 AM and 1:00 PM* – ✓
36. *Hit Cursor Down and Enter to select "1:00 PM"* – ✓
37. *Check Checkboxes Classical and Blues* – ✓

38. *Select Favorite Color "Blue"* – ✓
39. Fill in Description:
    1. *Triple Click Name field* – ✓
    2. *Check that complete Name is selected* – ✓
    3. *Press Ctrl+C* – ✓
    4. *Click into Description Field* – ✓
    5. *Paste with Ctrl+V* – ✓
    6. *Check that Text is the same as in Name field* – ✓
40. Done.

Overall Impressions

Selenium

First of all it gives a good feeling that the developers are aware of Selenium as Web-Test tool as can be derived from their blog. This raises the chance that the ExtJS developers will improve ExtJS in that way that Selenium can handle it more easily. Better propagation of component IDs to the rendered component will be a first major step to take.

The installation of Selenium as Firefox plugin was a piece of cake. And recording worked quite well despite not recorded mouse and keyboard events. Another problem is the required right click to record checks. Workaround is to add the checks manually later on in the test. If only the focus change is a problem the test steps to restore the previous state (e. g. refocus to another field) need to be deleted after recording.

Programming in the large (generating frameworks etc.) will require to export the tests as e. g. Java Test Case. I doubt that it is possible to get the tests back into the IDE then which poses a problem when you need to debug the tests.

Windmill

The installation of Windmill felt very heavy-weight to me as I had none of the required packages preinstalled like for example Python. This also means that preparing hosts for automatic test execution will take some time and produces much overhead if you want to run your tests on multiple machines. Perhaps the setup can be put in batch scripts.

Opening local files in Windmill is not possible for an unknown reason. For a real test scenario this is not relevant as you won't provide your web application in a single HTML file but with some server behind to interact with.

The corrupted layout during the first tests (reproducible and also checked that this did not occur outside of Windmill) gave a bad impression. Twittering this problem I got immediate response from Adam Christian, Co-Founder and Developer of Windmill. So the developers are caring and listening.

QF-Test

First of all: QF-Test was the only tool which passed the Capture & Replay-Test: Once recorded it was possible to replay the tests without any difficulty. The reason why this worked seems to be that fault tolerant approach of QF-Test identifying components. And: QF-Test seems to detect the generic IDs: It simply ignores the trailing numbers of the IDs.

During refactoring it is very easy to determine how components are identified. For example if one of the attributes must match, if attributes are matched against a regular expression and such.

The whole component tree can be found as tree in QF-Test which gives a good overview of the structures without further need for plugins like Firebug. In addition it serves as independent description of the component setup – independent from the actual tests which just reference the components by the QF-Test-IDs (which not necessarily need to match the HTML-element-IDs). Within QF-Test it was easy to detect that e. g. the company selector button is embedded in a div-node which contains the component ID as part of its ID. This way identification of the company selector button is very robust.

QF-Test was the only tool which replayed keyboard and mouse events without problems. And it is the only tool which provides a check which component is currently focused using the proxy class DomNode and its method `hasFocus()`.

The only problem I observed was that nodes sometimes got added multiple times to the component tree map in QF-Test. One reason seems to be that ExtJS often changes the class attribute e. g. when hovering above the elements. The map can be refactored that such classes are ignored – but it is nearly not feasible to clean the map after the complete test run. Thus it is recommended to record step by step and clean the map afterwards. This will ensure that you will have a robust and easy to maintain test afterwards.

QF-Test has many other advantages like a great debugger with step-by-step debugging, automatic screenshots on errors, good to read reports and screenshot-comparison. This is why we already use it for a long time for tests of our Java Swing-applications.

From all test tools evaluated here QF-Test is the only commercial one. For Web-Tests you currently have to pay (rough values, please ask QFS for exact pricing):

- EUR 1300 per developer license plus EUR 250 to receive updates and premium support
- EUR 900 for a runtime license (sufficient for nightly automatic tests) plus EUR 150 for updates/support

If you already use QF-Test e. g. for Java Swing-Tests you can upgrade (again only rough prices):

- EUR 350 per developer license upgrade plus additional EUR 75 updates/support

Conclusion

Web applications may be the most challenging task for GUI-Test-Automation. Not only that they render differently in different browsers to get around the quirks of the browsers: Also the element structure is more complex than those of for example a Java Swing application.

So it is even more important to find a tool which helps you to get around these obstacles. Taking the free tools out there Selenium seems to be a good choice. Especially because the ExtJS developers are aware of it. Recording is easy and it is a plus that the recorded test cases can be exported to any language (like Java for example). This ensures that your developers can export it to their language they favor most. Of course this also means that the main development and debugging will be done outside of the IDE.

If you are willing to pay for a test tool I recommend QF-Test. For CoreMedia it has an additional benefit as we also develop Swing applications and thus can stick to one tool for all. The support is great and fast and QF-Test made great improvements during the past years which makes the future look very promising.

About the author

Mark Michaelis is Software Engineer Quality Assurance at CoreMedia since 2005. He collected a lot of experiences doing tests with debugging and developing automated GUI-Tests. These experiences include

- the pain debugging badly written GUI-Tests,
- working with GUI-Test-Tools which produce random results when tests are replayed and
- the benefit and good feedback for a well written test-framework for GUI-Tests

Mark Michaelis has great knowledge of the tool QF-Test and minor knowledge of other Web-Test-Tools.

See also

- GUI Testing – Tool Requirements and Test Guidelines; Blog Post by Mark Michaelis, 04/22/2009
- QF-Test: Fallstudie CoreMedia Content Management System; Mark Michaelis 06/27/2007 (Original English Version at Facebook)
- QF-Test: Evaluationsbericht; Mark Michaelis January 2006
- Boon and Bane of GUI Test Automation; Mark Michaelis in Testing Experience 04/2008
- Sahi – Web Automation and Test Tool
- Windmill Testing Framework
- Selenium web application testing system
- QFS – developers of QF-Test
- HFS – HTTP File Server – a simple webserver