Project:
A DSL for browser automation

Browser Automation

In your successful job of Web application architect, you have often to perform repetitive tasks with your browser: operations like logging into systems, filling forms, navigating to specific locations, etc... Your colleagues (especially in the testing department) have the same problem, and some of them are not experienced programmers. You are asked to improve your company productivity by implementing an external DSL for browser automation.

By this DSL you want to be able to express procedures like:

- open a browser window (e.g., Firefox)
- go on “http://campus.mines-nantes.fr”
- click on the link “Log in”
- click on the button with text “Cliquez ici ou sur le logo C'zam pour vous identifier”
- fill the text field “username” with “mtisi08”
- fill the text field “password” with “12345”
- select the checkbox named “warn”
- click on the button with text “LOGIN”
- go to the url « https://campusneo.mines-nantes.fr/campus/»
- verify that the link “Consulter le profil” contains the string “Massimo Tisi”

Besides basic navigations like the previous example, the DSL should allow users to read information from a point of the Web page, and use it later in an operation.

Selenium WebDriver

You were starting developing your Java program to do all that, when you discovered that a Java library that performs exactly the operations you need is already freely available. It's Selenium http://docs.seleniumhq.org/.

Selenium 2.0 provides a very simple API to automate browser execution, WebDriver. A short but useful documentation is provided here: http://docs.seleniumhq.org/docs/03_webdriver.jsp

Tasks

Design and implement the browser automation DSL as an external language, representing a flow of browser operations. Selenium greatly simplifies the task of implementing the semantics of your language, but there is still a lot of work to do:

- design a textual syntax for your DSL and generate a textual editor using Xtext (with syntax highlighting, code assistance, autocompletion, outline, ...)
- implement a compiler to Java, to generate an executable program to deploy in the machines of your colleagues.

The final product of your work is going to be a set of Eclipse plug-ins that, when added to an Eclipse installation, will allow users to create programs in your DSL by using a textual syntax, and to execute them.

Test 1
* open a browser window
* go to the url "http://www.imt-atlantique.fr/fr"
* verify that the page contains a link "Toutes les actualités"

Test 2
* open a browser window
* go to the url " http://www.imt-atlantique.fr/fr"
* click on the link "Toutes les actualités"
* verify that the page contains a link "Accueil"

Test 3
* open a browser window
* go to the url "http://www.imt-atlantique.fr/fr"
* click on the link "Toutes les actualités"
* click on the image with alt property "Accueil"
* verify that the page contains a link "Toutes les actualités"

Test 4
* open a browser window
* go to the url "http://www.imt-atlantique.fr/fr/rechercher"
* insert "Donald Trump" in the search field
* click on the button "Appliquer les filtres"
* verify that the page contains the text "Aucun résultat ne correspond à votre recherche"

Test 5
* open a browser window
* go to the url "http://www.imt-atlantique.fr/fr/formation/trouver-ma-formation"
* uncheck all checkboxes
* check the checkboxes "Anglais", "A domicile" et "Temps plein"
* click on the button "Appliquer les critères"
* verify that the page contains the text "No found course matching your criteria."

Test 6
* open a browser window
* go to the url "http://www.imt-atlantique.fr/fr/rechercher"
* insert "2007" in the search field
* choose the option "Le mois dernier" in the combobox
* click on the button "Appliquer les filtres"
* verify that the page contains the text "Aucun résultat ne correspond à votre recherche"

Test 7
* open a browser window
* go to the url "http://www.imt-atlantique.fr/fr"
* click on the title of the first post (class="actu_home_ctner_inner_cell1_titre") in the page
* verify that the new page contains the title of the previously clicked post

Test 8
* open a browser window
* go to the url "http://www.imt-atlantique.fr/fr"
* read the title of the second post (class="actu_home_ctner_inner_cell1_titre") in the page and the url it points to
* go to the url "http://www.imt-atlantique.fr/fr/rechercher"
* paste the title in the search field
* click on the button "Appliquer les filtres"
* verify that the page contains a link to the previous url