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 a 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 “connexion”
- click on the picture “C'zam”
- fill the text field “username” with “mtisi08”
- fill the text field “password” with “12345”
- select the checkbox « Prévenez-moi avant de me connecter à d'autres services »
- click on the button “connexion”
- go to the url « https://nte.gemtech.fr/campus/course/view.php?id=1415 »

Besides basic navigations like the previous example, the DSL should allow users to :

- define loops, e.g. do/while;
- apply operations on collections of page elements, e.g. “select all the checkboxes of the page”;
- define conditional flow, e.g. if/then/else;
- read information from a point of the Web page, and use it later in an operation ;
- define parametrical subprocedures and call them from other procedures.

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/.

Surprisingly you also read on Wikipedia that ''Selenium 2.0 aims to provide a basic set of building blocks from which developers can create their own Domain-specific Language'' (http://en.wikipedia.org/wiki/Selenium_(software)).

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 process language in Java, 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 the abstract syntax of your language,
- implement an interpreter for executing your language from the development environment,
- implement a compiler to Java, to generate an executable jar to deploy in the machines of your colleagues,
- design a textual syntax for your DSL and generate a textual editor using tools like TCS or Xtext,
  - with syntax highlighting, code assistance, autocompletion, outline, ...

- design a graphical syntax for your DSL and create a graphical editor in Eclipse.

Schedule

Session 1: January 24th
Session 2: February 21st
Submission: March 1st