Web Application Testing Automation Framework

Automation Test Engine (ATE)

– An Open Source Project Powered by Selenium –

Peidong Hu @ Montreal Prot QA
peidong@bigtester.com
hup@vaniercollege.qc.ca
514 803 6688

Jun Yang
wpbxwpbx@gmail.com
5149622387

Why market is choosing Selenium

Feature	**QTP (UFT)**	**Selenium (WebDriver)**
Programming Lang	VB script	Java, C#, Ruby, Python, Perl, PHP, Javascript
Browsers Supported	Chrome, IE, FireFox	Chrome, IE, FireFox, Opera, HTMLUnit
Environment Supported	Windows	Windown, Linux, Unix, OSX, Others
Mobile Support	UFT Mobile	Android, iPhone/iPad, Blackberry etc.
Infrastructure	HP QC	Eclipse, Maven/Ant, Jenkins, TestNG, SVN
Reports	HP QC	Jenkins
Software Cost	Expensive (License + Renewal)	Free

Montreal Inf

Why market is choosing Selenium

Feature	**QTP**	**Selenium**
Coding Experience to start	Fair	Good
Script Creation Time	Less or High	High
Code Maintainability	Depends on script design	Good
Overall Cost	**Software Cost + QA developer + Code Maintenance**	**QA developer + Code maintenance**
Job openings On Indeed.com	6	10

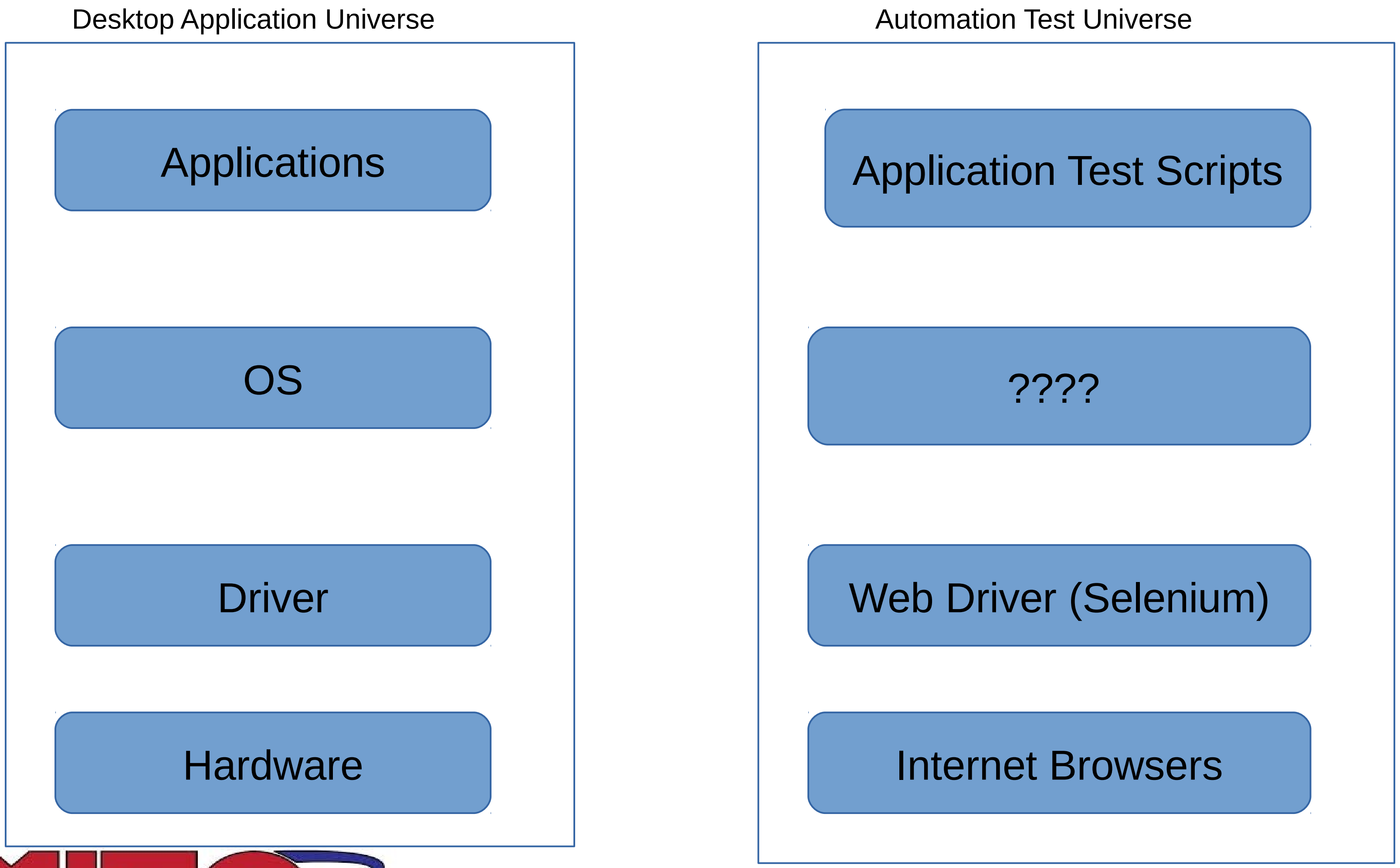
Montreal Information Technology Club

Why does it cost so much to build Good scripts

- Consideration
  - Maintainability
  - Flexibility
  - Re-usability
  - Portability
  - Extensibility
- Key
  - Modularity

Montreal Information Technology Club

Reason of high cost to build good testing scripts



Montreal Information Technology Club

Automation Test Engine

- Fill the gap between Driver (WebDriver) and Application (Test Script)
- Created all the modules required by 'good design'
- Programming Languages are not necessary to build application test scripts
- Only HTML, SQL and XML skills required
- Improve script readability
- All the other good-abilities in 'good design'

MITC
Montreal Information Technology Club

Basic modules in ATE design

1) Test Projects

2) Test Suites

3) Test Data

4) Test Report

5) Test Log

6) Test Cases

7) Test Steps

8) Test Configuration and tear down

9) Test Services (Test Step Sets)

10) Conditional Steps/Services

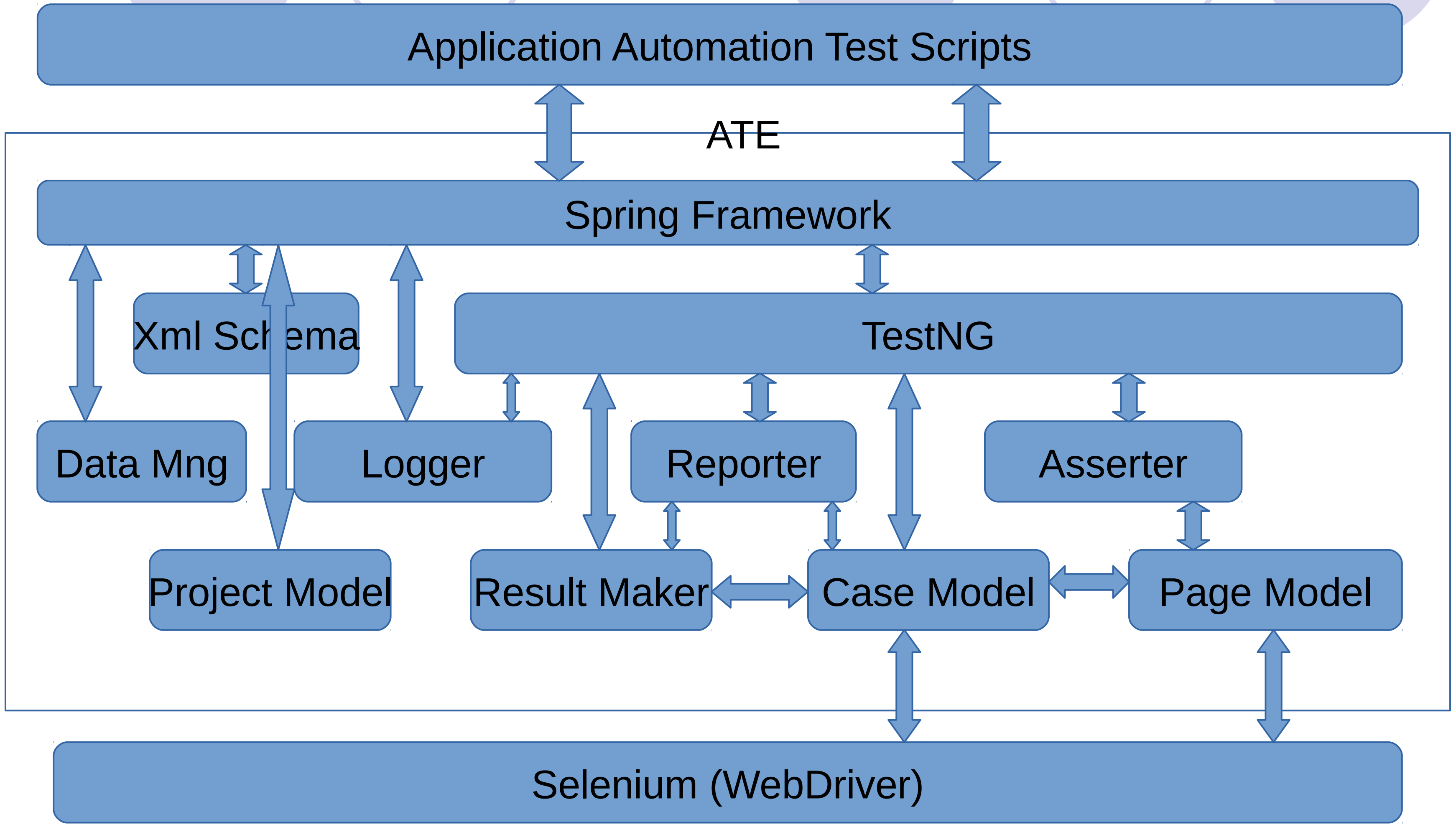
11) Loop-able Steps/Services

12) Exception Handling

13) Assertion

Montreal Information Technology Club

ATE Architecture



Montreal Information Technology Club

Examples with ATE – scripting

```
<ate:elementStep id="step1" stepName="InputUserName" stepDescription="step1 type user name"
        targetStep="false" myWebElement="mywebelem1" >
    <property name="expectedResultAsserter">
        <list>
             <ref bean="LoginPagePropertyAsserter" />
             <ref bean="LoginElementExistenceAsserter" />
         </list>
    </property>
</ate:elementStep>

<ate:myWebElement id="mywebelem1">
        <constructor-arg ref="elemfind1" />
        <constructor-arg ref="elemaction1" />
</ate:myWebElement>

<ate:elementFindById id="elemfind1" findByValue="modlgn_username" />

<ate:sendKeysAction id="elemaction1" dataValue="userNameValue" />
```

Montreal Information Technology Club

Examples with ATE – reporting

```
<Step name="InputUserName" index="2">
      <StepDescription>
             <![CDATA[step1 type user name ==> peidonghu1x]]>
      </StepDescription>
      <StepResult>
<![CDATA[
Page_Property_Correctness ==> HIGH ==> Page_Title ==> My Page Title ==> null ==> null ==> NotCorrect ==>  PAGEPROPERTYNOTCORRECT
Page_Element_Existence ==> HIGH ==> ID ==> logo1 ==>  ==> NotExist ==>  ==> PAGEELEMENTNOTEXIST
Page_Element_Existence ==> MEDIUM ==> ID ==> banner ==>  ==> Exist ==>  ==> PAGEELEMENTEXIST
Page_Element_Existence ==> HIGH ==> ID ==> search_phu ==>  ==> Exist ==>  ==> PAGEELEMENTEXIST
]]>
      </StepResult>
</Step> <!-- input user name -->
```

Montreal Information Technology Club

Exmaples with ATE – execution

Command line execution

1) java -jar ate.jar testproject/testproject.xml

Or

2) maven clean test

Montreal Information Technology Club

Examples without ATE – Scripting

```
WebElement myDynamicElement = (new WebDriverWait(driver,
 10)).until(ExpectedConditions.presenceOfElementLocated(By.name("login")));

 // Find the text input element by its name

 WebElement element = driver.findElement(By.name("username"));

 // Enter something to search for

 element.sendKeys("Cheese!");

 String pageTitle = driver.getTitle();

 WebElement logo = driver.findElement(By.id("logo"));

 WebElement searchbox = driver.findElement(By.id("userSearch"));

 Assert.assertTrue(pageTitle.equals(newTitle));

 Assert.assertTure(!logo.isEmpty());

 Assert.assertTure(!searchBox.isEmpty());
```

MITC
Montreal Information Technology Club

Examples without ATE – Reporting

Need additional hundreds lines of code to be able to support step level report

Examples without ATE – Execution

Need many lines of code to be able to run it in command line and with maven

Montreal Information Technology Club

Important modules that ATE implements

1) Test Projects
2) Test Suites
3) Test Data stores in XML (ElementFind data) and Database (Input data, result data.)
4) Test Report Enhancement including Test Step level report, Element level report
5) Test Log including Application level Log and System low level log
6) Test Cases
7) Test Steps
8) Test Configuration (in beta2)
9) Test Services (Test Step Sets) (in beta2)
10) Conditional Steps/Services (in beta2)
11) Loop-able Steps/Services (in beta2)
12) Exception Handling
13) Assertion

MITC
Montreal Information Technology Club

ATE using the following technologies/libraries

1) Java
2) Test Projects, Test Suites, Test Cases - Spring Framework
3) Test Data stores in XML (ElementFind data) - DBUnit
4) Test Data stores in Database – Spring JPA, Hibernate, HyperSQLDB, DBUnit
5) Test Report – AOP programming, TestNG, Jenkins and customized plugins
6) Test Log – Spring AOP programming and LogBack
7) PageObject, Test Steps – Spring Framework
8) Test Configuration and Teardown (in beta2)
9) Test Services (Test Step Sets) (in beta2)
10) Conditional Steps/Services (in beta2)
11) Loop-able Steps/Services (in beta2)
12) Exception Handling – AOP programming and Problomatic chain Library
13) Assertion – TestNG and AOP programming
14) Extension feature – Spring Framework
15) Eclipse and Spring IDE development environment
16) Maven build/deployment and Jenkins CI and reporting system
17) Github source code control

MITC
Montreal Information Technology Club

Future of ATE

1) Flexible engine can run with different testing driver
2) Integrate Web Service interface testing
3) GUI test case composer – eclipse plugin
4) Intelligent manual test case converter

Montreal Information Technology Club

ATE is in Alpha release

https://github.com/bigtester/automation-test-engine

And its relevant projects at

https://github.com/bigtester/

Section 2 – ATE Live Demo

Live Demo in Eclipse, Live Demo in Maven and Live Demo in Jenkins

Selenium only Live Demo (without ATE)

0) Selenium Demo

1) ATE and Selenium Element Find

2) ATE and Selenium Element Action

3) ATE other concepts (xml element introduction)

- a) Test Project, Suite, TestCase
- b) Test Step, Asserter
- c) InputData and ResultData
- d) Data.xml file
- e) Test Report and Test Log
- f) Extensibility

Option: standalone Java executable demo (local version)

Montreal Information Technology Club