Performance analysis of various User Interface Test Automation for Silverlight applications

Appasami. G and Suresh Joseph. K

Abstract - Silverlight applications are used to create platform independent, browser independent interactive and attractive web applications with Dot net supporting languages. User Interface (UI) Test Automation is mainly used to reduce the time, cost and manual work. There are several methods are available like keyboard driven, mouse driven for UI Test Automation. In this paper we made performance analysis of various set of UI Test Automation for Silverlight applications. UI Test Automation is more difficult for web based applications when compared with windows based applications. UI accessibility and automation are very difficult for Silverlight applications. In this paper we analyzed different types of UI Test automation with different kinds of input data and output data. In this paper we analyzed the performance of various input data for various UI test automation.

Index Terms - User Interface, Test Automation, Automation Peer and Silverlight Applications.

I. INTRODUCTION

Silverlight is Microsoft's new cross browser or delivering richer interactive applications to users over the web. Silverlight 2.0 is Microsoft's second release of the Silverlight platform [18][19]. Silverlight 2's biggest change from Silverlight 1.0 is the inclusion of a compact version of the .NET Framework, complete with the .NET Framework 3.0 Common Language Runtime. By adding .NET to Silverlight, Microsoft makes it easy for .NET developers to reuse their existing programming skills, collaborate with designers, and quickly create rich applications for the Web. One of the key benefits of Silverlight 2 is that it can execute any .NET language, including C# and VB.NET [17][18]. Silverlight 2 ships with a "lightweight" version of the full .NET Framework, which features, among other classes, extensible controls, XML Web Services, networking components, and LINQ APIs. This class library is a subset of the .NET Framework's Base Class Library, which enables the Silverlight plug-in to be a fast and small download. In addition to the .NET Framework classes, Silverlight 2 also ships with a subset of the WPF UI programming model, including support for shapes, documents, media, and WPF animation objects [15][18][19].

Testing these kinds of new applications with existing old testing methods are ridicules [1][3][4]. We have to test new applications developed by new technology and also reduce test time and cost.

Microsoft is developed Silverlight and provides accessibility classes [15][21]. Automation Peer UI Test approach is best to do test automation for Silverlight applications [19][22].

II. TYPES OF UI TEST AUTOMATION

Software test automation has evolved through several generations of tools and techniques. Test Automation is started from Character user Interface to till now. It can be classified based on testing strategy [12][24]. They are:

A. Capture/playback tools record the actions of a tester in a manual test, and allow tests to be run unattended for many hours each day, greatly increasing test productivity and eliminating the mind-numbing repetition of manual testing. However, even small changes to the software under test require that the test be recorded manually again [4][11]. Therefore this first generation of tools is not efficient or scalable [12][16].

B. Scripting is a form of programming in computer languages specifically developed for software test automation, alleviates many issues with capture/ playback tools. However, the developers of these scripts must be highly technical and specialized programmers who work in isolation from the testers actually performing the tests. In addition, scripts are best suited for GUI testing and don't lend themselves to embedded, batch, or other forms of systems. Finally, as changes to the software under test require complex changes to the associated automation scripts, maintenance of ever-larger libraries of automation scripts becomes an overwhelming challenge [4][11][12][17].

C. Data-driven testing is often considered separately as an important development in test automation [6]. This approach simply but powerfully separates the automation script from the data to be input and expected back from the software under test. This allows the data to be prepared by test engineers without relying on automation engineers, and vastly increases the possible variations and amounts of data that can be used in software testing. This breaking down of the problem into two pieces is very powerful. While this approach greatly extends the usefulness of scripted test automation, the huge maintenance chores required of the automation programming staff remain [11][13][16].

D. Keyword-based test automation breaks work down even farther, in an advanced, structured and elegant approach [6]. This reduces the cost and time of test design, automation, and execution by allowing all members of a testing team to
focus on what they do best. Using this method, non-technical testers and business analysts can develop executable test automation using “keywords” that represent actions recognizable to end users, such as “login”, while automation engineers devote their energy to coding the low level steps that make up those actions, such as “click”, “find text box A in window B”, “enter User Name”, etc. Keyword-based test design can actually begin based on documents developed by business analysts or the marketing department, before the final details are known. As the test automation process proceeds, bottlenecks are removed and the expensive time of highly trained professionals is used effectively. Organizing test design and test automation with the keyword framework eliminates time [11][14].

E. UI Test Automation peer classes provide a powerful framework for organizing test design, test Automation and Test execution by AutomationID properties [20][21]. Using these tests can pass values to controls as well as they can invoke events for that controls, an automation engineer can focus on automating actions as individual building-blocks that can be combined in any order to design a test. Non-technical test engineers and business analysts can then define their tests as a [Test class], and execute their tests automatically without creating any additional code. AutomationPeer test design takes place in a spreadsheet, with actions listed consecutively in a clear well organized sequence of Actions, test data and any necessary GUI interface information are stored in their own spreadsheets, from which they can be called by the main test module. Tests are then executed from right within the spreadsheet, using custom-built Automation Test classes. AutomationPeer classes allow testing teams to create a much more effective test automation framework, overcoming the limitations of other methods. AutomationPeer Testing significantly reduces the maintenance burden by allowing users to define their tests at the business process level. Rather than defining tests as a series of interactions with the UI, test designers can define tests as a series of business actions. It will be the job of the automation engineer to update the actions affected by the UI changes, and this update will only need to be made in only one place, rather than in multiple test scripts. Automation Peer Accessibility classes method is the best choice for Dot Net web based Silverlight Applications [11][20][25].

III. PERFORMANCE ANALYSIS

For any kind of software Applications Performance analysis is very important. From the performance analysis and comparative study one can easily choose their choice based on the analysis report. In this paper we took input data from like Text, Excel, XML, MS-SQL and Oracle. UI Automation Report will be produced in Text file, Excel sheet and XML. Silverlight is Microsoft’s new cross browser or delivering richer interactive applications to users over the web. Silverlight 2.0 is Microsoft’s second release of the Silverlight platform. Now a day’s web sites are developed with this Silverlight for best visualization and attraction.

A. Input Data

We maintained records in various databases like Text, Excel, XML, MS-SQL and Oracle. In each data bases we maintain the same records for this performance analysis. The UI Test Automation is initially started with 5000 records, then 10000, 15000 and 20000. Finally the average case is taken for our analysis.

1) Text File

Maintaining records in text file is very easy, but security is very less. Records are stored in a row and columns are separated by commas. File size is very small. Searching a particular record based some values are not so easy.

2) Excel Sheet

In Excel, Data Entry is very easy and look and feel is very nice. Data records are stored in a row like a matrix. Excel provides little bit security when compared with Text file. Searching a value and sorting on a particular column are easy.

3) XML Data

XML Files approach is Easy to use in any application. Data update is very simple and can be accessed via standard XML APIs. XML Data is universally understandable, portable and Best suitable for internet. Data retrieval is very fast. Now XML has advanced locking, synchronization and concurrency control mechanism. If we want ease of use and ability to transfer data easily between multiple applications than XML files approach seems right one [26][30].

- Web publishing: XML allows you to create interactive pages, allows the customer to customize those pages, and makes creating e-commerce applications more intuitive. With XML, you store the data once and then render that content for different viewers or devices based on style sheet processing using an XSL/XSLT processor.
- Web searching and automating Web tasks: XML defines the type of information contained in a document, making it easier to return useful results when searching the Web.
- General applications: XML provides a standard method to access information, making it easier for applications and devices of all kinds to use, store, transmit, and display data.
• E-business applications: XML implementations make electronic data interchange (EDI) more accessible for information interchange, business-to-business transactions, and business-to-consumer transactions.

• Metadata applications: XML makes it easier to express metadata (Unified Modeling Language design models or user interface properties, for example) in a portable, reusable format.

• Pervasive computing: XML provides portable and structured information types for display on pervasive (wireless) computing devices such as PDAs, cellular phones, and others.

```xml
<!-- file: Customer.xml -->
<?xml version="1.0"?>
<!-- our XML-document describes a Customer Details -->
<Customer>
  <Record>
    <Name>Appasami</Name>
    <Address>109-Pk st Kalapet</Address>
    <City>Pondicherry</City>
    <Country>India</Country>
    <Phone>9786554175</Phone>
    <Email>appas_9y@yahoo.com</Email>
  </Record>
  <Record>
    <Name>John</Name>
    <Address>12-mouni Rue</Address>
    <City>Pondicherry</City>
    <Country>India</Country>
    <Phone>9345197987</Phone>
    <Email>johnmtch@gmail.com</Email>
  </Record>
  <Record>
    <Name>Appasami</Name>
    <Address>40-MGRoad</Address>
    <City>Beijing</City>
    <Country>China</Country>
    <Phone>9245762106</Phone>
    <Email>aravin_28@yahoo.com</Email>
  </Record>
</Customer>
```

Figure 3. XML data for Input

4) **MS-SQL**

MS-SQL is a Microsoft server database with the Following Features:

• Security of data from unauthorized user.

• Compatible for all dot net Applications.

• Transparent Data Encryption. The ability to encrypt an entire database.

• Auditing. Monitoring of data access.

• Data Compression. Fact Table size reduction and improved performance.

• Resource Governor. Restrict users or groups from consuming high levels or resources.

• Hot Plug CPU. Add CPUs on the fly.

• Performance Studio. Collection of performance monitoring tools.

• Installation improvements. Disk images and service pack uninstall options.

• Dynamic Development. New ADO and Visual Studio options as well as Dot Net 3.

• Entity Data Services. Line Of Business (LOB) framework and Entity Query Language (eSQL)

• LINQ. Development query language for access multiple types of data such as SQL and XML.

• Data Synchronizing. Development of frequently disconnected applications.

5) **ORACLE**

Oracle has many advantages and features that makes it popular and thereby makes it as the world’s largest enterprise software company. Oracle comes with new versions with new features implemented in new version while the features of earlier versions still being maintained [28][29]. Some features of Oracle listed below.

• Role Based Security and fast adoption standards.

• Backward compatibility and Automatic Diagnostic Repository.

• Oracle is used for almost all large applications.

• Ease of customization and installation.

• Scalability and Performance (Concurrency, Consistency, Locking Mechanisms and Portability)

• Manageability (Self managing database, OEM, SQL*Plus, ASM, Scheduler, Resource Manager)

• High availability, Backup and Recovery

• Business Intelligence (Data Warehousing, OLAP, Data mining, Partitioning)

• Data integrity/Triggers

• Information Integration Features (Distributes SQL, Oracle Streams)

```
<table>
<thead>
<tr>
<th>Name</th>
<th>Address</th>
<th>City</th>
<th>Count</th>
<th>Phone</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>Appasami</td>
<td>109-Pk st Kalapet</td>
<td>Pondicherry</td>
<td></td>
<td>9786554175</td>
<td><a href="mailto:appas_9y@yahoo.com">appas_9y@yahoo.com</a></td>
</tr>
<tr>
<td>John</td>
<td>12-mouni Rue</td>
<td>Pondicherry</td>
<td></td>
<td>9345197987</td>
<td><a href="mailto:johnmtch@gmail.com">johnmtch@gmail.com</a></td>
</tr>
<tr>
<td>Aravintha</td>
<td>40-MGRoad</td>
<td>Pondicherry</td>
<td></td>
<td>9245762106</td>
<td><a href="mailto:aravin_28@yahoo.com">aravin_28@yahoo.com</a></td>
</tr>
</tbody>
</table>
```

Figure 5. Oracle data for Input

B. **UI Test Automation**

UI Test Automation is very essential for Software industries to reduce test time, cost and man power. Now-a-days web applications are developed by new technologies like Silverlight, JAVADEX, FLEX, etc. Silverlight is new .NET technology to develop rich interactive Internet applications [11][19][22][23]. Testing these kinds of applications are not so easy to test, especially the User
interface test automation is very difficult. Dot net 3.5 Framework Provides Automation Peer Accessibility class, using this we can develop test automation for Silverlight applications [1][3][11].

In this paper, we took five kinds of input data like Text, Excel, XML, MS-SQL and Oracle. For UI Test automation we took Automation Peer Accessibility classes method, this is the best choice for Silverlight Applications [20][25][27]. The out put of UI Test Automaton is generated in Excel sheet with both input and output data. UI Test Automation program will undergo slight modification if any controls changed, so common reusable code is maintained for all UI Test Automation [3][5][7][8]. The design of UI Test Automation with Input data types, Test Method types and Output report types is shown in figure 6 [10][19][20].

Silverlight testing must collect information about all the controls available on that testing page then it form a tree with UI elements as nodes. The particular element is searched in Depth first searching technique. There should not any loop with in the search tree to search a particular element in search tree. Finally the value is passed to the particular element by UI Automation.

C. Output Report

The out put of UI Test Automaton can be easily updated in Excel sheet with both input and output data. The output report for UI Automation of Silverlight applications is as shown in the figure.

IV. RESULTS AND DISCUSSIONS

The performance analysis is done in terms of security, reliability, portability, Integrity and processing time. We maintained records in various databases like Text, Excel, XML, MS-SQL, Oracle. In each data bases we maintain the same records for this performance analysis. The UI Test Automation is initially started with 5000 records, then 10000, 15000 and 20000. Finally the average case is taken for our analysis. The Approximately estimated time in nano Seconds for 5000, 10000, 15000 and 20000 records is given as shown in the table 1.
XML (Time in ns) | 100 | 600 | 1400 | 1600 | 2100 | 1425.0  
MS-SQL (Time in ns) | 400 | 900 | 1450 | 1900 | 2400 | 1662.5  
Oracle (Time in ns) | 450 | 950 | 1000 | 1950 | 2450 | 1587.5

<table>
<thead>
<tr>
<th>DBs</th>
<th>Data Entry</th>
<th>Space for data</th>
<th>Secu rity</th>
<th>Input Time</th>
<th>Report Output</th>
<th>Perfor mance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text</td>
<td>Easy</td>
<td>Small</td>
<td>No</td>
<td>Less</td>
<td>Fast</td>
<td>Better</td>
</tr>
<tr>
<td>Excel</td>
<td>Easy</td>
<td>Small</td>
<td>Less</td>
<td>Less</td>
<td>Fast</td>
<td>Better</td>
</tr>
<tr>
<td>XML</td>
<td>Moderate</td>
<td>Small</td>
<td>Less</td>
<td>Less</td>
<td>Fast</td>
<td>Best</td>
</tr>
<tr>
<td>MSSQL</td>
<td>Difficult</td>
<td>Large</td>
<td>More</td>
<td>More</td>
<td>Slow</td>
<td>Good</td>
</tr>
<tr>
<td>Oracle</td>
<td>Difficult</td>
<td>Large</td>
<td>More</td>
<td>More</td>
<td>Slow</td>
<td>Good</td>
</tr>
</tbody>
</table>

Table 2. Performance comparison table for various DBs.

V. CONCLUSION

There are several methods for UI automation, but AutomationPeer method is best for Silverlight applications. MS-SQL and Oracle Database provide more security, but it takes more access time when compared with XML data. Developer can choose any database according to their needs, but XML is best input data for Silverlight UI Automation, because entering data is easy and Transferring Time is also less. For Security purpose MS-SQL and Oracle are best. Without considering security, XML data is best input for Silverlight UI Automation. AutomationPeer Method of UI Test Automation is best for Silverlight UI Test Automation. To generate output report, Excel is best one when compared with others like Text, XML, MS-SQL and Oracle.

REFERENCES


[22] http://www.jeff.wilcox.name/2008/03/silverlight2-unit-testing


Appasami G was born in Pondicherry, India in 1980. He received his Master of Science degree in Mathematics and Master of Computer Applications degree from Pondicherry University, Pondicherry, India; Pursuing Master of Technology in Computer Science and Engineering from Department of Computer Science, Pondicherry University, Pondicherry, India. Currently he is working as Research Assistant in Department of Computer Science, Pondicherry University, Pondicherry, India. His Area of interests includes image processing, neural network and web technology.

Suresh Joseph K was born in Tamil Nadu, India in 1978 received his Bachelor of Engineering degree from Bharathiyar University and Master of Engineering (Computer Science and Engineering), from University of Madras, Chennai, India, in 2002. Since 2006, he has been an Assistant Professor at the Department of Computer Science, Pondicherry University, and Pondicherry, India. His research interests include Image processing and Software Engineering.