

Comparative Performance Analysis of UI Test Automation from Various Databases for New Web Applications

G Appasami¹ K.Suresh Joseph²

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.

Keywords : User Interface, Test Automation, Automation Peer and Silverlight Applications, Performance Analysis.

1. 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].

¹Dept. Of Comp. Science, Pondicherry Univ.
Email : appas_9g@yahoo.com

²Lect, Dept. of CS, Pondicherry Univ.
Email : sureshjosephk@yahoo.co.in

2. USER INTERFACE MARKUP LANGUAGES FOR A WEB APPLICATION

Ajax, XUL, XAML, Flex, platforms are emerging, each one with its advantages and its drawbacks, and the choice is posed since one wants to build a Web application, a RIA (Rich Internet Application) having the same interface with attractive and interactive user interfaces.. We listed all widely used (for graphical user interface) markup languages, and we have to choose the best one for our needs.

A. GladeXML

GladeXML is the XML format used by the Glade Interface Designer. It creates forms that can then be used in conjunction with the libglade library using GTK+. Glade provides a graphical interface development environment in the model of Visual Studio, C++ Builder and so on [31].

B. XUL

XUL (XML User Interface Language), pronounced *zool*, is an user interface markup language developed to support Mozilla, Firefox and Mozilla Thunderbird. XUL reuses many existing standards and technologies, including CSS, JavaScript, DTD, RDF and XPCOM[32].

C. Open Laszlo and Flash

Open laszlo is a free application based on a markup language named LZX. On the client-side, it produces Flash code which can be runs thanks to a plugin rather commonly installed, and it produces also DHTML code by Java script [33].

D. MXML

MXML is an XML markup language introduced by the Macromedia in 2004. Apart the design of user interface, it can also be used in conjunction with ActionScript to implement complex business logic. alternative from Macromedia to XUL and XAML is used by Web applications in conjunction with developments dedicated for the interaction with the browser. It has a development tool named Flex [34].

E. XAML

XAML is like XUL an XML-based language of description of interface language for silverlight. But unlike XUL it is at start intended by Microsoft to be a means of creating rich Internet applications. Just as XUL is recognized by Firefox, XAML is recognized by Internet Explorer, with the advantage of a larger number of users. He thus supplements the universal .NET platform, with a language of interface. It is provided natively with Vista and is used for the graphic interface of Vista [19].

Table 1 Shows the comparison of most popular UI Languages with Accessibility and security levels.

Table 1 : Shows The Comparison of Most Popular UI Languages

	Year	Development	Runtime	Processing	Languages	Requirements	Accessibility	Security
GladeXML GNOME	1998	Glade IDE	GTK+	Compiled	C, C++, C#	XML	Easy	Low
XUL Mozilla	1998	Text editor	XULRunner	Interpreted	ECMAScript, C++	CSS, DTD, RDF, XPath, XPCOM	Easy	Low
OpenLaszlo Laszlo Systems	2003	Text editor	Flash Player	Compiled	ECMAScript	CSS, XPath	Medium	Medium
MXML Macromedia	2004	Flex Builder	Flash Player / Apollo	Compiled	ActionScript	CSS	Medium	Medium
XAML Microsoft Silverlight	2006	Microsoft Interactive Designer	WinFX / Silverlight	Compiled	.NET languages / JavaScript	XPath, .Net	Difficult	High

UI Test Automation Types :

1. For Easy accessibility with low security the UI Test Automation will be easy.
2. For moderate accessibility with medium security the UI Test Automation will be moderate.
3. For y with high security the UI Test Automation will be very difficult.

So Silverlight UI Test Automation is very difficult because XAML accessibility is Difficult and has high security

3. 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 testers 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 testers 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. Automation Peer 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. Automation Peer classes allow testing teams to create a much more effective test automation framework, overcoming the limitations of other methods. Automation Peer 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].

4. 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.

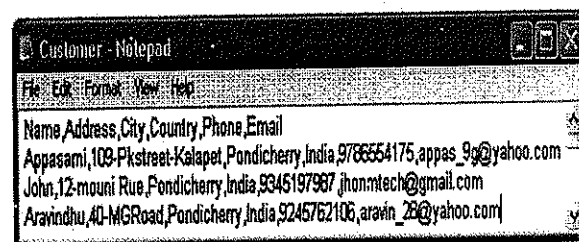


Figure 1 : Text Data File For Input

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.

Comparative Performance analysis of UI Test Automation from various Databases for new web applications

	A	B	C	D	E	F
1	Name	Address	City	Country	Phone	Email
2	Appasami	109-Pk st Kalapet	Pondicherry	India	9786554175	appas_9@yahoo.com
3	John	12-mouni Rue	Pondicherry	India	9345197987	jhonmtech@gmail.com
4	Aravindhu	40-MGRoad	Beijing	China	9245762106	aravin_28@yahoo.com

Figure 2 : Excel Sheet Data For Input

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 is 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

```

<!-- 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_9@yahoo.com</Email>
  </Record>
  <Record>
    <Name>John</Name>
    <Address>12-mouni Rue</Address>
    <City>Pondicherry</City>
    <Country>India</Country>
    <Phone>9345197987</Phone>
    <Email>jhonmtech@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 of 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.

- 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)

Oracle SQL PLUS					
Select * from Customer					
Name	Address	City	Country	Phone	Email
Appasami	109-Pk street	Kalapat	Pondicherry	India	9786554175 appas_9@yahoo.com
John	12-mounl Rue		Pondicherry	India	9345197387 jhonmtech@gmail.com
Aravindhu	40-MGRoad		Pondicherry	India	9245762106 aravin_2@yahoo.com

Figure 5 : Oracle Data For Input

Name	Address	City	Country	Phone	Email
Appasami	109-pk st kalapat	Pondicherry	India	9786554175	appas_9@yahoo...
John	12-mounl Rue	pondicherry	India	9345197387	jhonmtech@gma...
aravindhu	40-MGRoad	Beijing	China	9245762106	aravin_2@yahoo...

Figure 4 : MS-SQL Data For Input

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.

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, JAVAFX, 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 output 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].

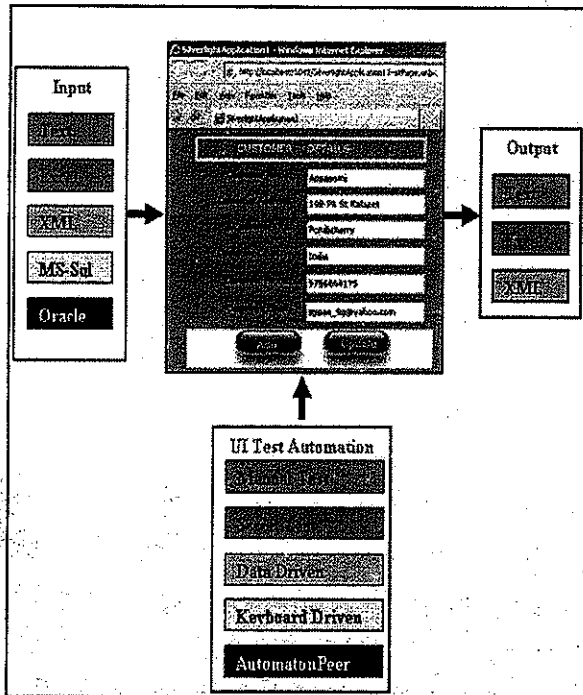


Figure 6 : UI Test Automation With Input Data Types, Test Method Types And Output Report Types.

The sample code for UI Test automation using Automation peer method is given below [24][25][30]. Initially one process object is created and then we get the handle from the running browser. Tree walker object is created to find a particular element in the web page [20]. Using Silverlight objects we can set values to text box and then we click Buttons like add and update. These things are done by our program code test method [23][24][25].

```
//Test1.cs
[TestMethod]
public void TestMethod1()
{
    Process process =
    System.Diagnostics.Process.GetProcessesByName("iexplore").First();
    BrowserInstance =
    System.Windows.Automation.AutomationElement.FromHandle
    (process.MainWindowHandle);

    TreeWalker tw1 = new TreeWalker(new PropertyCondition
    (AutomationElement.AutomationIdProperty, "TextBox"));
    AutomationElement searchTextBox =
    tw1.GetFirstChild(browserInstance);
    TreeWalker tw2 = new TreeWalker(new PropertyCondition
    (AutomationElement.AutomationIdProperty, "Button"));
    AutomationElement searchTextBox = tw2.GetFirstChild(browserInstance);
    SilverlightApp app = ActiveBrowser.silverlightAppas[0];
    app.FindName<TextBox>("TBox1").Text="Appasami";
    app.FindName<TextBox>("TBox2").Text="109-pk street Kalapet";
    app.FindName<TextBox>("TBox3").Text="Pondicherry";
    app.FindName<TextBox>("TBox4").Text="India";
    app.FindName<TextBox>("TBox5").Text="9786554175";
    app.FindName<TextBox>("TBox6").Text="appas_9g@yahoo.com";
    app.FindName<Button>("ADD").User.Click();
}
}
```

Figure 7 : Sample Code For UI Test Automation

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.

	A	B	C	D	E	F	G
1	Appasami	109-Pk st	Pondichar	India	9786554175	appas_9g@yahoo.com	Successfully Updated
2	John	12-msumi	Pondicher	India	9345197967	johnmdech@gmail.com	Successfully Updated
3	Aaravindho	40-MGRoad	Baijing	China	9245762106	aravin_2@yahoo.com	Successfully Updated

Figure 8 : Excel Output Report

5. 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.

Table 1 : Different Database Records and Their UI Automation Processing Time

Databases	Time input	Time for UI Automation (NS)				AVG
		5000	10000	15000	20000	
No. of Records		5000	10000	15000	20000	12500
Text (Time in ns)	250	750	1300	1750	2250	1512.5
Excel (Time in ns)	300	800	1100	1800	2300	1500.0
XML (Time in ns)	100	600	1400	1600	2100	1425.0
MSSQL (Time in ns)	400	900	1450	1900	2400	1662.5
Oracle (Time in ns)	450	950	1000	1950	2450	1587.5

The graph is plotted for the average case. In this Experiment the average number of records is taken as 12500 and the same records are maintained in all data files like text file, Excel sheet, XML Data, MS-SQL and Oracle table. The graph shows that the UI Test Automation time Using XML is very less when compared with others. The overall UI Test Automation processing time for each database is shown on the top of each column as shown in the bar chart.

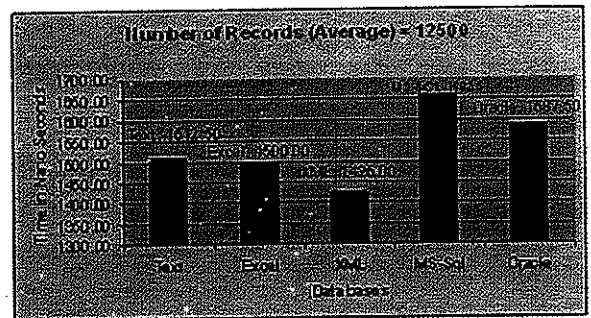


Figure 9 : Chart For Databases And Processing Time With 12500 Records

In this performance analysis, we considered the Easy of data entry, Space required for data, input process and Output report generation time for UI Test Automation. Even though databases like MS-SQL and Oracle provides more security, XML data set is best for UI Test Automation. Usually managing text files are very easy. So entering data in text files and excel sheet are easy, but these kinds of data does not provide security. Developer, tester and User can choose Database according to their needs. But for UI Test automation tester XML is best because of its good performance [2][9][11][12]. The performance comparisons are shown in Table2.

Table 2 : Performance Comparison Table for various DBs

DBs	Data Entry	Space for data	Security	Input Time	Report Output	Performance
Text	Easy	Small	No	Less	Fast	Better
Excel	Easy	Small	Less	Less	Fast	Better
XML	Moderate	Small	Less	Less	Fast	Best
MSSQL	Difficult	Large	More	More	Slow	Good
Oracle	Difficult	Large	More	More	Slow	Good

6. 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. Automation Peer 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

- [1] Anna Derezsinska and Tomasz Malek, "Experiences in Testing Automation of a Family of Functional- and GUI-similar Programs", International Journal of Computer Science & Applications, Technomathematics Research Foundation, Vol. 4, No. 1, PP. 13–26, June 2007.
- [2] Q. Xie and A. M. Memon, "Designing and comparing automated test oracles for GUI-based software applications", ACM Transactions on Software Engineering and Methodology, Vol. 16, No. 1, PP. 4-es, Feb 2007.
- [3] Zhu Xiaochun, Zhou Bo, Li Juefeng and Gao Qiu, "A test automation solution on GUI functional test", IEEE Conference on Software Maintenance, 6(2): PP: 1413-1418, July 2008.
- [4] A. M. Memon, "An event-flow model of GUI-based applications for testing", IEEE conference on Software Testing, Verification and Reliability, Vol. 17, No. 3, PP. 137-157, September 2007.
- [5] A. M. Memon, M. E. Pollack and M. L. Soffa, "Hierarchical GUI test case generation using automated planning", IEEE Transactions on Software Engineering, PP : 144–155, Feb. 2001.
- [6] White. L, Almezen. H, "Generating test cases for GUI responsibilities using complete interaction sequences", Proceedings of the International Symposium on Software Reliability Engineering, 8–11, IEEE Computer Society Press: Piscataway, NJ, 2000, 110–121, October 2000.
- [7] White. L, Almezen. H, Alzeidi. N, "User-based testing of GUI sequences and their interaction", Proceedings on Software Reliability Engineering, IEEE Computer Society Press: Piscataway, NJ, 2001, 54–63, 8–11, November 2001.
- [8] Memon A, Nagarajan. A, Xie. Q, "Automating regression testing for evolving GUI software", Journal of Software Maintenance and Evolution: Research and Practice, 17(1):27–64, 2005.
- [9] A. M. Memon and Q. Xie, "Studying the fault-detection effectiveness of GUI test cases for rapidly evolving software", IEEE Transactions on Software Engineering, 31(10):884–896, 2005.
- [10] Q. Xie and A. M. Memon, "Designing and comparing automated test oracles for GUI-based software Applications", ACM Trans on Software Engineering and Methodology, 16(1):4, 2007.
- [11] Appasami.G and Suresh Joseph. K, "Automation Peer – User Interface Test Automation for Silverlight Applications", IETECH Journal of Advanced Computations, India, April 2009.
- [12] Fewster, "Software Test Automation", Addison Wesley, 1999.
- [13] Kanglin Li and Mengqi Wu, "Effective GUI Test Automation: Developing an Automated GUI Testing Tool", SYBEX Inc., 2005.
- [14] Kanglin Li and Menqi Wu, "Effective Software Test Automation: Developing an Automated Software Testing Tool", ISBN:0782143202 Sybex Inc, 2004.

- [15] Tom Arnold, Dominic Hopton, Andy Leonard and Mike Frost, "Professional Software Testing with Visual Studio® 2005 Team System", Wiley Publishing, Inc. 2007
- [16] Elfriede Dustin, "Effective Software testing", Pearson Education Inc., 2003.
- [17] Brad Dayley and Lisa DaNae Dayley, "Silverlight™ 2 Bible", Wiley Publishing, Inc., 2008.
- [18] Matthew MacDonald, "Silverlight 2 Visual Essentials", Firstpress, 2008.
- [19] <http://www.silverlight.net>
- [20] <http://code.msdn.microsoft.com/silverlightut>
- [21] <http://silverlight.net/learn/tutorials/controls.aspx>
- [22] <http://www.jeff.wilcox.name/2008/03/silverlight2-unit-testing>
- [23] [http://msdn.microsoft.com/en-us/library/cc645045\(VS.95\).aspx](http://msdn.microsoft.com/en-us/library/cc645045(VS.95).aspx)
- [24] <http://weblogs.asp.net/scottgu/archive/2008/04/02/unit-testing-with-silverlight.aspx>
- [25] http://dotnetslackers.com/Patterns_and_Practices/UI_Automation_Testing_with_UIA_Veify.aspx
- [26] <http://publib.boulder.ibm.com/iserics/v5r1/ic2924/index.htm?info/rzak1/rzaklntroadvantages.htm>
- [27] <http://artoftestinc.blogspot.com/2008/08/automated-testing-of-silverlight.html>
- [28] http://it.toolbox.com/wiki/index.php/Oracle_11G_New_Features
- [29] <http://www.adp-gmbh.ch/ora/misc/features.html>
- [30] <http://www.codeguru.com/forum/archive/index.php/t-204260.html>
- [31] <http://www.xml.com/pub/a/2001/09/05/xforms.html>
- [32] <http://www.xul.fr/en-xml-ajax.html>

- [33] <http://www.openlaszlo.org> – open Laszlo http://www.adobe.com/...coding_with_mxml_and_actionscript/

Author's Biography



Appasami. G received his Master of Science degree in Mathematics, Master of Computer Applications Degree and Master of Technology degree in Computer Science and Engineering from Pondicherry University, Pondicherry, India. Currently he is working as a lecturer in the Department of Computer Science and Engineering, Dr. Pauls Engineering College, Villupuram, Tamil Nadu, India. His Area of interests includes image processing and web technology.



Nakkeeran. R received his Master of Science, Master of Computer Applications Degree and Master of Engineering (Computer Science and Engineering) from Anna University, Chennai, India. Since 2007, he has been working as a Professor in the Department of Computer Science and Engineering, Dr. Pauls Engineering College, Villupuram, Tamil Nadu, India. His research interests includes web technology, DBMS and Software Engineering.



Suresh Joseph. K 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 includes Image processing and Software Engineering.