Model-Based Testing is Being Used at Microsoft

Y.Sherly Priscilla Assistant. Professor, NethalaTulasiRaju Assistant. Professor,
K.Rambabu Assistant. Professor

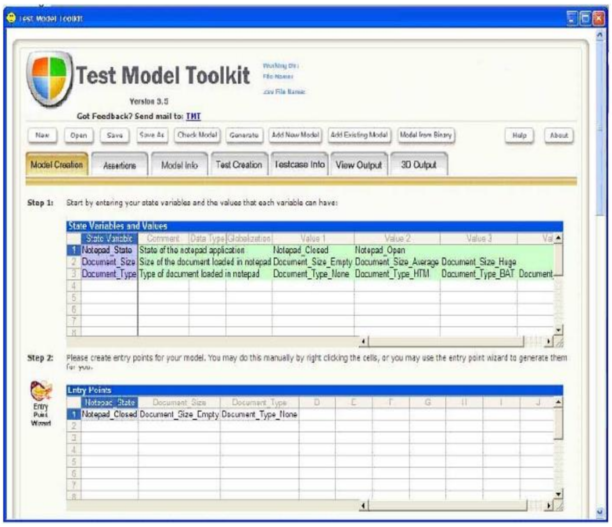
Abstract

For its Trustworthy Computing [4] initiative, Microsoft has been investigating potential methods to increase reliability. The model-based testing method is gaining popularity and attention. There is a need for more robust and flexible modelling due to the widespread usage of a Finite State Machine modelling tool (TMT). The Amsel/T testing system and the Abstract State Machine Language (Amsel) are now under investigation by many product teams. They are both more effective at discovering issues early on, during the design and definition phases, and at testing live systems to cover more structural code.

Keywords:*formal testing methods, choosing which tests to run, and creating test cases automatically.*

Introduction

Most software testing is done via black box testing, which involves observing the program's behaviour from the outside. It wasn't long ago that many testing were conducted by hand. As the software and environment requirements for running Microsoft products have become more complex, the need of having an easier test design and administration has grown. Even though many testers utilise models in their brains or on paper, Model-Based Testing isn't entirely usable due to its lack of automation. The folks working on Internet Explorer devised a method to model things using FSMs. in English



Step 1: Start by entering your state variables and the values that each variable can have:

State Variables and Values

	State Variable	Comment	Data Type	Globalization	Value 1	Value 2	Value 3	Va
1	Notepad_State	State of the notepad application			Notepad_Closed	Notepad_Open		
2	Document_Size	Size of the document loaded in notepad			Document_Size_Empty	Document_Size_Average	Document_Size_Huge	
3	Document_Type	Type of document loaded in notepad			Document_Type_None	Document_Type_HTM	Document_Type_BAT	Document
4								
5								
6								
7								
8								

Step 2: Please create entry points for your model. You may do this manually by right clicking the cells, or you may use the entry point wizard to generate them for you.

Entry Points

	Notepad_State	Document_Size	Document_Type	D	E	F	G	H	I	J
1	Notepad_Closed	Document_Size_Empty	Document_Type_None							
2										
3										
4										
5										
6										
7										
8										

team and is now really used a lot. A group of product workers saw Abstract State Machine Language (ASML) from the Foundations of Software Engineering (FSE) team at the MSR. This more complicated method has been met with the most enthusiasm by test groups when regular FSMs fail. Indigo is a set of.NET technologies for making and controlling networked computers. It is the connectivity backbone of Longhorn. FSE and the Indigo development team have been working together to use Amsel and Amsel/T. A number of Microsoft groups have looked into Amsel and Amsel/T.

Test Model Toolkit

The Test represent Toolkit is a Finite State Machine-based application where state tables are used to represent the product being tested. TMT employs a state table with three columns: initial state, action, and final state. Testers with no programming knowledge may be taught TMT and the FSM technique.

Generating test cases with test queries

Test questions are used to look into the model and make test cases. Because each query is a question you want to ask your model, it can be used to make a huge number of test cases. The six types of queries that can now be used on TMT are You can make random test cases for any number of runs with experimental. By guided, all test cases with up to a certain number of steps are made. quickest Path finds the quickest way between the locations of where your test case starts and where it ends. All quickest Pathways finds and gives you the quickest routes between any two test case start and end places that you can think of. Random Shortest Path or Shortest Path You Choose One question can be made from two using this method. Choose the starting state where you want to find the quickest route, and the question will make a set of test cases that are either random or directed. You can use these questions to make test cases that will put your product through its paces in the most dangerous areas.

Assertions

Assertions are a sort of basic model checking that is supported by TMT. Assertions are non-negotiable claims made by the tester about their domain. Not only are assertions helpful for checking the integrity of the model itself, but they may also be used to guarantee the integrity of the standard the model was developed from.

Automation

Assertions are a sort of basic model checking that is supported by TMT. Assertions are non-negotiable claims made by the tester about their domain. Not only are assertions helpful for checking the integrity of the model itself, but they may also be used to guarantee the integrity of the standard the model was developed from.

Automation

XML Test Cases (etc) are the files that are generated by TMT while running a test. In addition, an automated template is produced by TMT. The test scripts are written by the tester who will execute them. This was the first tool that really assisted in automating test designs as opposed to merely running tests automatically.

Impact

TMT is used by at least twenty Microsoft groups. Up to 88% of the work that was done by hand was taken away. It would have taken eight weeks of work to make test cases by hand, but BizTalk did it automatically. It went over up to 50% more of the code. One Shell team was able to increase the number of test cases from 75 to 2000 in just two weeks. As you work on making a model, many problems with the specification and application become clear. [10] TMT was seen as the best way to test software inside Microsoft in the early spring of 2001.

Amsel:

The FSE team helped pilot Amsel in many different teams in Microsoft with several resulting papers some of which appear on their web site. [1].

What is Amsel?

Amsel is a language for writing instructions that can be carried out. It is based on the idea of Abstract State Machines. This new version is already built into Microsoft Word and Visual Studio.NET. XML and Word are used to write literate specs. And it works with every.NET language out there. You can run.NET assemblies you make with Amsel from the command line, connect them to other.NET assemblies, or package them as COM components. Specifications made in Amsel are very clear and can be used to talk about any type of computer system. A standard from Amsel makes it possible for program managers, coders, and testers to all speak the same language. One of the best things about it is that an Amsel standard can be run. That's why it's helpful to do this before writing the system's code. Asking yourself "Does it do everything you planned it to do?" and "How well does it work?" can help you figure out your design. How do the different parts work together? In what ways does it hurt?

Adoption

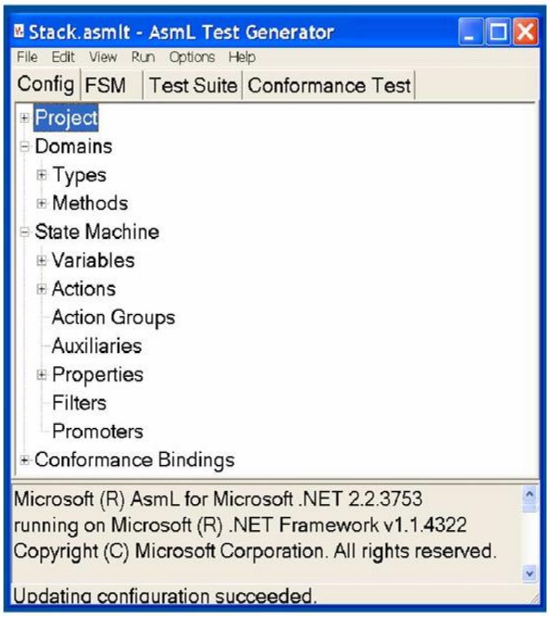
Although there are few exceptional exceptions, very few testers are able to teach themselves Amsel. Every year, the Indigo testing team received a two-day training session to bring them up to speed. Not all pupils were eager to adopt the idea, but many did. However, the development of a model is expensive, and current techniques in the business do not effectively gauge either problem avoidance or quality enhancement. As a result, testers face an uphill battle when trying to earn credit for averting errors by, say, asking clarifying questions and fixing mistakes in the specification well before any code or test cases are written.

Impact

Indigo is founded on a set of Web Services (WS) standards that have been developed by experts in the field. The Indigo test team used FSE's modellingtools to aid in the development of these standards. It took the writers months of design and architectural meetings to discover the same twenty problems that we discovered in a week of modelling the WS-Routing standard [6]. Six problems were discovered while modelling the WS-Policy specification [7], and they were all resolved. We discovered recent problems with the updated specification. Many modelling questions about the WS Atomic Transactions standard [10] arose from this work. There were so many problems with the original specification while modelling the new Passport login protocol in Amsel that it was redone from scratch.

Amsel/T:

The Amsel Test Tool [2] is a product of Amsel. The Amsel Test Tool allows you to quickly and easily construct tests from an Amsel model. It is possible to generate test sequences and parameters. You may use the tool to ensure your implementation is up to line with the requirements by running your Amsel model alongside your code. The tool provides a basic automation interface, making it easy to integrate into an existing testing framework. Since the same data for things like Methods and Actions must be presented in many methods, it is clear that this first-generation tool was designed as a collection of Researchers areas.



```
Microsoft (R) AsmL for Microsoft .NET 2.2.3753
running on Microsoft (R) .NET Framework v1.1.4322
Copyright (C) Microsoft Corporation. All rights reserved.
```

Updating configuration succeeded.

Adoption:

Small/T helped evaluators get more out of their time spent developing models. Improvements to requirements, test case generation, and the availability of an oracle are all now possible. This was really a research prototype; hence the price was exorbitant. The learning curve was high, documentation was sometimes lacking, and topics were often articulated in unfamiliar (research) words. Adoption was often achievable only because of the FSE team's training, advising, mentoring, and coaching.

Impact:

One incompatibility problem was discovered in WS-Routing testing of Web Service Extension 1.0 [8], following conventional testing. Indigo's first WS-Policy implementation has thirteen flaws, which were discovered via the usage of Amsel/T parameter generation to construct different policies. In the early implementation of WS-Atomic Transaction in Indigo, more than three very sneaky problems were discovered.

Model Based Testing: Research to Practice

Scientists thrive on the challenge of addressing intractable issues. Unfortunately, it is often the simplest issues that have been addressed that become a roadblock for practitioners when trying to implement new developments. True testers use automated regressions, free of user interface (UI) or prompts. As part of a bug report to development, genuine testers may recreate any situation at will, save it, and boil it down to its barest essentials [5].

Futures:

Model Based Testing is used by a lot of Microsoft testers. These testers talk about the method's benefits and how to get started at workshops and in trade magazines and blogs. [11] As TMT grows into an Integrated Test Environment, it gets better FSM Modelling, N-wise parameter combos, and test tools that work together. The FSE team is using Indigo Tests' feedback to make Amsel and testing tools that are easier to use (with shorter learning curves and more features for testers with more experience), easier to fix, and have been tested and tested again and again.

Futures:

Although basic models like Cause-and-Effect Graphs and Finite State Machines should already be in widespread usage for Model Based Testing, they are not. Most testers still believe the tools are either too difficult or too pricey for their needs. The majority of focus groups still need constant evangelizing in the form of case studies and advising. In addition, when businesses start using MBT, they often discover that the basic models are not robust enough to deal with their specific issues. It is necessary to make it easy for testers to go from scratching the surface with elementary models to managing large swaths of their testing domain with complex models that support model verification and execution.

Acknowledgements

Many people deserve credit for this work, including but not limited to Sara Wong and the rest of the Indigo testing team, as well as the members of Microsoft Research's Foundations of Software Engineering (FSE) group (whose members have included Colin Campbell, Yuri Gurevich, Wolfgang Ryskamp, Wolfram Schulte, Nikolai Tillmann, and Margus Vanes).

References

[1] Abstract state machine language is available at http://research.microsoft.com/fse/asml/, which is maintained by Microsoft Research. [2] Visit http://research.microsoft.com/fse/AsmL/doc/AsmLTester.html to get the Amsel Test Generator utility developed by Microsoft Research. the third Read M. Corning's "Confessions of a Modelling Bigot: Parts III-IV" and ASP Today from 2002 and 2003 (http://www.asptoday.com/content.asp?id=2043).

Fourthly, a trustworthy computing URL for dependable computing Located at http://www.microsoft.com/mscorp/twc/reliability/default.mspx, it is available to you. A. Zeller and R. Hildebrandt published a paper titled "Simplifying and Isolating Failure-Inducing Input" in the February 2002 issue of the IEEE Transactions on Software Engineering, volume 28, issue 2, pages 183–200. http://msdn.microsoft.com/library/default.asp [6] Web Services Routing Protocol (WS-Routing), H.F. Nielsen and S. Thatte, October 23, 2001 [7] Digital document originating from D. Box about the Leaf Services Policy Framework (WSPolicy), dated May 28, 2003. Visit http://msdn.microsoft.com/ws/2002/12/Policy/ to see Microsoft's WS Policy page. provide more details on the policy. [8] WSE 1.0 SP1 from Microsoft.NET: Web

Services Enhancements You may get it by going to www.microsoft.com/downloads/details.aspx.

[9] WS-Atomic Transaction, the Web Services Atomic Transaction described by L.F. Cabrera et al. in September 2003 (http://msdn.microsoft.com/library/en-us/dnglobspec/html/wsat.asp).Electronic Notes in Theoretical Computer Science 111, Stobie, 2005, 12. 5–12 Examine Internet Explorer: How It Went From Opponent to Top Browser, by J. Taylor, Star West 2002.

the eleventh H. Robinson's 1999 Star West article "Finite State Model-Based Testing on a Shoestring" is available online at http://www.model-based-testing.org/shoestring.htm.