

Using Automation to Generate Automated Software Testing – Reducing the Human Interaction for Testing of DOD Systems

ABSTRACT-----

We specialize in automated testing solutions for the Department of Defense. One recent testing improvement initiative is the establishment of a task force to improve development test and evaluation. A “Memorandum for Chairman, Defense Science Board” with the subject “Terms of Reference – Defense Science Board (DSB) Task Force on Development Test and Evaluation (DT&E)”, states that “approximately 50% of programs entering Initial Operational Test and Evaluation (IOT&E) in recent years have not been evaluated as Operationally Effective or Operationally Suitable.” Because of this memorandum, dated 2007, it was requested that “DSB establish a task force to examine T&E roles and responsibilities, policy and practices, and recommend changes that may contribute to improved success in IOT&E along with quicker delivery of improved capability and sustainability to Warfighters.”

This presentation talks about how we have used automated software testing to improve quality while decreasing cost.

BIOGRAPHY-----

Scott Bindas

Software Development Manager
Innovative Defense Technologies

Scott Bindas is the software development manager at Innovative Defense Technologies (IDT) (<http://www.idtus.com>). The focus of the software development team there is to deliver an automated software testing framework incorporating various open-source, vendor-provided, and IDT-developed software solutions that is scalable, reusable and expandable. This requirement led Scott and his team to focus on and implement automating the test automation.

Prior to joining IDT, Scott worked at a major defense contractor as a principal software engineer. In that role he was responsible for the software design, development, and testing of applications for Navy submarine programs. Scott holds a Bachelor of Science in Computer Engineering from the University of Massachusetts, Dartmouth.

Elfriede Dustin

Innovative Defense Technologies

Elfriede Dustin currently leads IDT’s (<http://www.idtus.com>) efforts in automated software testing research programs. Elfriede invented the Automated Testing Lifecycle Methodology (ATLM©) described in the book “Automated Software Testing;” which has served as a baseline for automated software testing implementation at various companies throughout the world. Together with IDT’s president Bernie Gauf and co-worker Thom Garrett at IDT, a company that specializes in automated testing solutions for the DOD, Elfriede wrote “Implementing Automated Software Testing,” Addison Wesley, March 2009, along with her books “Effective Software

Testing,” “Automated Software Testing,” “Quality Web Systems,” “The Art of Software Security Testing” and “SAP Testing for Managers.” All books have been translated into various languages and are sold throughout the world.

Elfriede presents at various conferences, most recently at Google’s Test Automation Conference (GTAC 2008) is also the conference chair for Verify 2009 (<http://verifyconference.com>), and has implemented software testing in commercial and Government organizations.



The MIT 2009 Information Quality Industry Symposium



**Using Automation to Generate
Automated Software Testing**
*Reducing the Human Interaction for
Testing of DOD Systems*



By Scott Bindas & Elfriede Dustin
Innovative Defense Technologies



The MIT 2009 Information Quality Industry Symposium



Using Automation to Generate Automated Software Testing

Agenda:

- Automated Software Testing (AST) Definition
- AST Research and Findings
- Applying our Findings at IDT - Automating our Test Automation
- Demo



The MIT 2009 Information Quality Industry Symposium



Definition

Automated Software Testing (AST):

Application and implementation of software technology throughout the entire Software Testing Lifecycle (STL) with the goal to improve STL efficiencies and effectiveness.

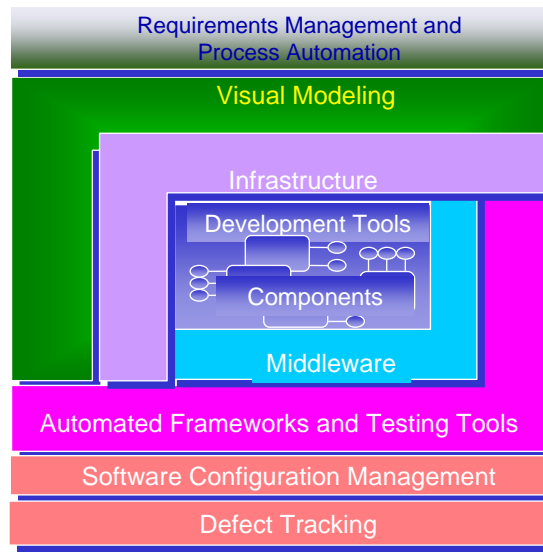


The MIT 2009 Information Quality Industry Symposium



Automated Software Testing Spans the Entire Lifecycle

- RM
- Modeling
- Middleware
- Infrastructure
- Defect Tracking
- Configuration Management
- Memory Leak Detectors
- Performance Testing tools
- Documentation tools
- more





The MIT 2009 Information Quality Industry Symposium



Software Testing Technologies - The Stepchild of Innovation

BusinessWeek Reports on "The World's Most Innovative Companies"
Spring, 2008 report:

http://www.businessweek.com/magazine/content/08_17/b4081061866744.htm?chan=magazine+channel_special+report

Focus is Research and Development (R&D)

Few Pay Attention to Research & Development & Test (R&D&T)

R&D has Resulted in High Tech Innovation

Our Ability to Test these Technologies has Not Kept Pace with Our Ability to Create Them

Innovation Does Not Seem to Consider Related Required Testing Technologies



The MIT 2009 Information Quality Industry Symposium



R&D & Testing Automate the Automation

Increasing software testing efficiencies

- For repeatability, scalability remove Human Interaction as much as possible

Automate the Test Automation

- Reuse and apply Hardware testing efficiencies
 - **hardware companies design a great deal of custom firmware/software to validate hardware designs/modules/systems, while still a lot of software testing is done manually**
- Borrow from Software Development R&D
- Reuse Open Source; existing components
- Automate components/parts integral to Automated Testing



The MIT 2009 Information Quality Industry Symposium



IDT AST R&D



The MIT 2009 Information Quality Industry Symposium



R&D & Testing

Hardware Testing

- Many automated test generators for manufacturing/ logic devices, circuit testing, etc., available i.e. SEI/ATG, uMaster, etc.
- Automated testing without human interaction is the norm for hardware testing



The MIT 2009 Information Quality Industry Symposium



Hardware vs. Software AT

How is Software Testing Different?

- Software Challenge:
 - Changing Requirements
 - GUI Interfaces
- Less Mass Production - Solution Flexibility Required – Problem Dependent

How is Hardware Testing the Same?

- Component Reuse
- Autonomous Computing; Self-testable components; self-configurable; self-healing
- Automate the Automated Testing (AT)



The MIT 2009 Information Quality Industry Symposium



R&D & Testing Component Reuse

Gartner and Redmond Developer Report on Need for Composite Applications

- Being able to capitalize on existing components; fragments; assemble these into custom built sources of information
- New user interfaces and transactions but behind the scenes composition of software resources into new patterns of applications
- Component reuse sites, such as www.koders.com and www.krugle.com
- Library concept of reusable components; checking in and out already tested code
- Reuse of already proven and tested components



The MIT 2009 Information Quality Industry Symposium



R&D&T

Autonomous Computing

- Autonomic computing (AC) is an approach to self-managed computing systems with a minimum of human interference. The term derives from the body's autonomic nervous system, which controls key functions without conscious awareness or involvement.
- Dynamic self-configuration, self-optimization, self-protection, and self-healing features of autonomic systems require that validation be an integral part of these systems.
 - <http://www.research.ibm.com/autonomic/>
 - DARPA's DASADA see http://www.alexander-egyed.com/publications/An_Externalized_Infrastructure_for_Self-Healing_Systems.html

Self-Testable Components

- <http://www2.computer.org/portal/web/csd/doi/10.1109/DSN.2001.941401>
- CONCAT – tool for self-testable classes



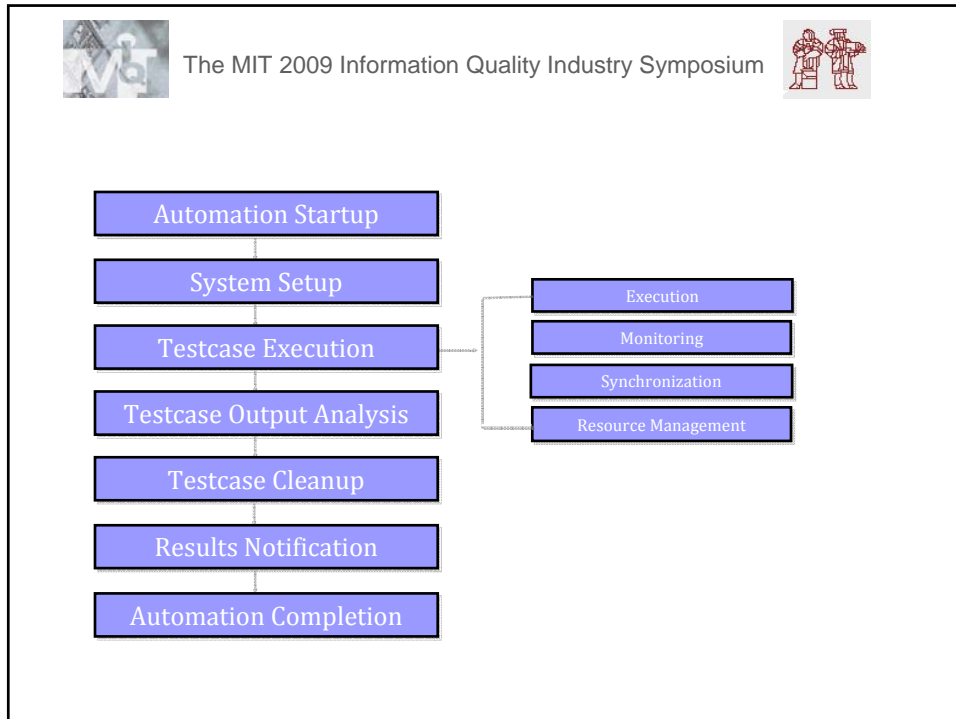
The MIT 2009 Information Quality Industry Symposium



R&D & Testing Automate the Automation

- Frameworks
- Test Data / Expected Results
- Various Input Files to Code Generation
- Automated Test Generators
- Automated Compare Utilities
- Log File Analyzers
- GUI Testing Tool Ideas
- New Tools...Other

- Working with NIST to Produce Additional Tools Such as Code Coverage Tools, etc...



The MIT 2009 Information Quality Industry Symposium

Automation Framework STAF/STAX


Software Testing Automation Framework (STAF) meets our needs
www.staf.sourceforge.net

With STAX you can automate and monitor your entire test environment


- System/product setup
- Testcase distribution
- Testcase execution
- Testcase results collection and analysis

STAX consists of

- A programming language designed for automation Programs written in this language are called jobs
- An execution engine (similar to an interpreter for other programming/scripting languages) which manages the execution and runtime behavior of the job
- A GUI application, called the STAX Monitor, which provides a dynamically updated view of your executing jobs
- A tool (STAXDoc) that is used to generate documentation for your STAX jobs



The MIT 2009 Information Quality Industry Symposium




End-to-End Automation with STAF & STAX

Automation Startup	Event, EventManager, Cron
System Setup	FS, Process
Testcase Execution	Process, Monitor, Log, Variable, Queue, ResPool
Test Case Output Analysis	Log, Process
Test Case Cleanup	Process
Results Notification	Email, HTTP
Automation Completion	


S
T
A
X

J
o
b

- <function>
- <paralle>
- <sequence>
- <process>
- <stafcmd>
- <testcase>
- <timer>
- <block>
- <loop>
- <message>
- <log>
- <import>
- <job>
- <iterate>
- <paralleliterate>
- <hold>
- <release>
- <terminate>



The MIT 2009 Information Quality Industry Symposium



Use of Automated Test Data Generators

GNU licensed- <http://generatedata.com>

NIST - Compares test data generators; have their own to tool FireEye
<http://csrc.nist.gov/groups/SNS/acts/documents/comparison-report.html>

Matlab - <http://www.mathworks.com/matlabcentral/>

Pairwise.org - <http://pairwise.org/tools.asp>

IBM Intelligent Test Case Handler - An Eclipse plug-in for creating test configuration data subject to coverage criteria
<http://alphaworks.ibm.com/tech/whitch/download>

Open Source - Test Vector Generator (TVG) provides a software tester with GUI to generate combinatorial Test Vectors based on the input-output relationship, N-way coverage or randomly.
<http://sourceforge.net/projects/tvg/>

Vendor provided tools: Datatect



The MIT 2009 Information Quality Industry Symposium



Automated Test Case Generation

IDT Conducted a Market Analysis of Automated Test Generation Products and We Found:

- Test generation based upon: UML model, Rules Checking, Algorithms, Test Descriptions
- Abundant Papers written by Industry and Academia
- Few Products actually built: Telelogic Rhapsody ATG, Teradyne TestMaster, ConformIq Test Generator, T-VEC
- **Limitation:** Tests have to be Generated within the Vendor Provided Software
- Decided to develop in house taking advantage of open source tools



The MIT 2009 Information Quality Industry Symposium



Automated Test Code Generation

Use of Various Input Types (such as IDL and C- Style Header Files) to Generate Test Code.

XML extracted from test case procedures.



The MIT 2009 Information Quality Industry Symposium



Automated Test Code Generation

Software Interface Test Code

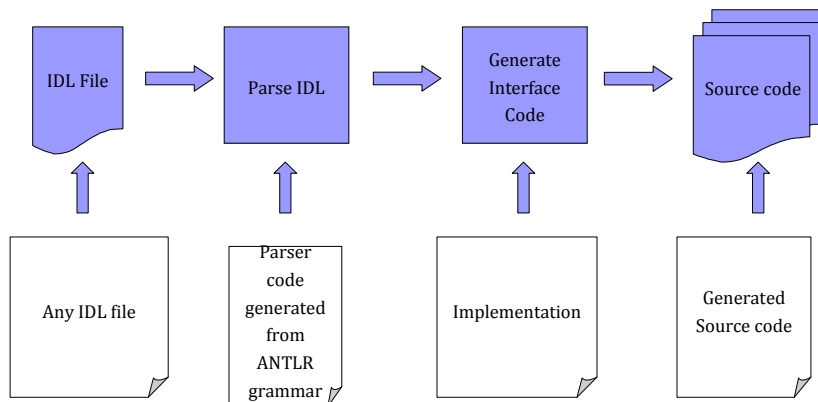
- A software interface could be a DDS, CORBA middleware or a simple Multicast layout.
- Generate software interface code from various input formats. For example, IDL, C-style headers, ...
- Generate code that glues the interface code to the test framework.



The MIT 2009 Information Quality Industry Symposium



Automated Test Code Generation





The MIT 2009 Information Quality Industry Symposium



Test Case Code Generation

Test Case Procedure Code

- Use csv parser to put test case in standard format
- Each step in the test case procedure has associated code generated to execute its behavior.
- A standardized test case format facilitates automated step procedure extraction.
- XML is used to define the extracted test step information as input into the autogen process.
- Stringtemplates provide for a common code set as well as project unique sets.

Test Case Builder



The MIT 2009 Information Quality Industry Symposium



Test Case Code Generation Tools

ANTLR - <http://www.antlr.org/>

- Works off of grammar files. Several common languages' grammar files available for download (eg. IDL & C).
- Generates lexical analyzer and parser functionality in Java. (Other languages supported as well.)

Stringtemplate - <http://www.stringtemplate.org/>

- Uses ANTLR. Developed by the same team.
- Provides pluggable templates. Code can be generated in multiple languages without having to modify the autogenerated code.

JAXB - <https://jaxb.dev.java.net/>

- XML to Java object binding based on XML schema.
- Generates Java classes used to interpret XML data.



The MIT 2009 Information Quality Industry Symposium



Automated Test Code Generation Benefits

When writing automation test code (code to verify System Under Test) consider auto generating code

- Provides predictable test software
- Scalable
- Portable
- Fix problems in one place
- Can reuse across projects
- Cuts down on development time per project, which makes automating tests more feasible
- Rough estimate: 70% Test case development reduction



The MIT 2009 Information Quality Industry Symposium



Use of Available Open Source Monitors and Log Parsers

<http://freshmeat.net/projects/logwatch/> - logwatch

Logwatch analyzes and reports on system logs. It is a customizable and pluggable log-monitoring system and will go through the logs for a given period of time and make a customizable report. It should work right out of the package on most systems.

Log File Monitor <http://sourceforge.net/projects/logfmon/>

Logfmon monitors a set of log files and processes messages based on a set of regexps. When a message matches, a command may be executed or the message may be ignored. Messages that are unmatched are batched together and emailed every so often.

LoFiMo - <http://sourceforge.net/projects/lofimo/>

Monitors logfiles and displays in realtime. Log messages may be formatted with different fonts and colors, contents can be parsed, reformatted, may even trigger actions playing sounds or executing shell commands. Log can be viewed remotely via webbrowser.



The MIT 2009 Information Quality Industry Symposium



Capture / Record & Playback

Much Research is focused on Capture/Playback tools

Automates actions of the test engineer:

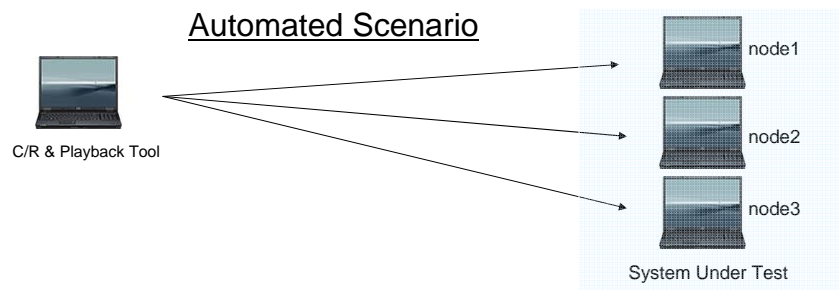
- During testing, the engineer uses the keyboard and mouse to perform some type of test or action.
- Testing tool captures all keystrokes and subsequent results, which are base-lined in an automated test script.
 - **IDT modified to generate code and library of reusable test functions**
- During test playback, scripts compare latest outputs with previous baseline.
- Provide for non-intrusive testing; i.e., they interact with the “application-under-test” as if the test tool was not involved.



The MIT 2009 Information Quality Industry Symposium



Capture / Record & Playback using VNC



1. C/R & Playback tool, allows for the automation of the previous “manual” tasks.
2. Tool acts as a VNC Client.
3. System Under Test comprised of multiple OS’s running in varied hardware.
4. Via X-Forwarding, not every node requires a running VNC Server.
5. Tools allow image comparison, object comparison, and text comparison
6. Tool may “playback” or execute its own comparison code.



The MIT 2009 Information Quality Industry Symposium



Capture / Record & Playback

Typical Automation Tool Script using VNC

```
Connect to node1
Find Image for menu X and Click
If (imageNotFound (menuX)){
  logError("menu X not found")
}

Connect to node2
Find expected window Y image for at most 10 seconds
If (imageNotFound (window Y)){
  logError("window Y not found")
}
Find image for drop down list Z and Click
Click new value from list
Find image for OK button on window Y and click

Connect to Node3
Find image of confirmation window
If (imageFound(confirmation window)){
  log("Confirmation Window Found")
}
Else{
  logError("Confirmation Window Not Found")
}
```



The MIT 2009 Information Quality Industry Symposium



Capture / Record & Playback

Methods of connection:

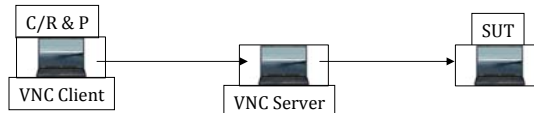
VNC Server Running on System Under Test

Fast/responsive but you now have process running on target hardware



X Forwarding/No VNC Server running on System Under Test

No test process (VNC Server) running on target hardware but this method can be potentially much slower with severe network bandwidth hit





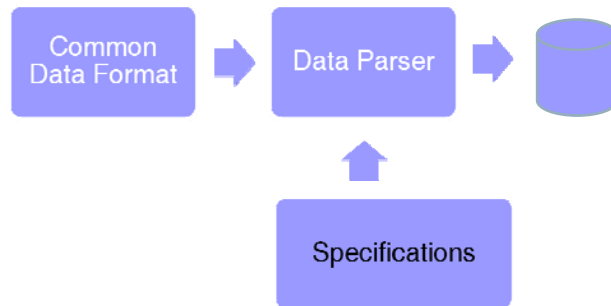
The MIT 2009 Information Quality Industry Symposium



Result Reporting

Reporting Overview:

- Keep result reporting as generic as possible.
- Keep independent of actual decision making
- Whether it's a Data Parser depicted below or an application reporting actual results from a test run



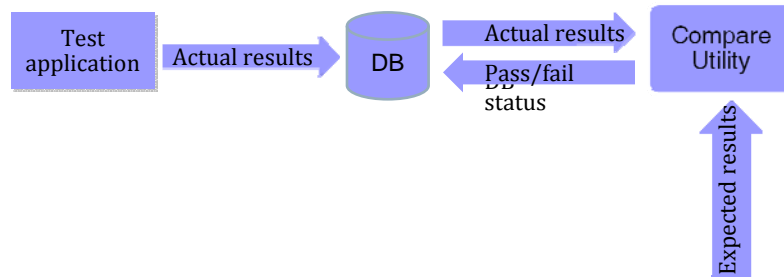
The MIT 2009 Information Quality Industry Symposium



Result Reporting

Puts comparison logic in utility

Only concerned with comparing Expected results (Text, integer etc...) with the actual results from the test run.





The MIT 2009 Information Quality Industry Symposium



Bug Reporting

Can fill in most of the information needed for a software trouble report

Provide summary and let user decide whether it's a bug

Use information gathered from the test run to provide detailed Trouble report

Will help developers recreate failures

Type of information

- Test case name and step number
- Requirement that failed
- Test data used
- Description of failure



The MIT 2009 Information Quality Industry Symposium



Demo

Demo



The MIT 2009 Information Quality Industry Symposium



Automation Testing Support Tools Used

Rich Client Platform

Used to build a test manager GUI with an Eclipse look-and-feel and the advantages of building on the Eclipse framework.

http://wiki.eclipse.org/index.php/Rich_Client_Platform

Postgres

An opensource database used to store test case data range from test step inputs to results comparisons.

<http://www.postgresql.org/>

Hibernate

An opensource framework for allowing data to be persisted in an object oriented fashion.

<http://www.hibernate.org/>

Apache POI

A Java API for accessing Microsoft formatted files. Useful in interacting with existing test case artifacts.

<http://poi.apache.org/>



The MIT 2009 Information Quality Industry Symposium



Key Automation Architecture Requirements

Support applications running on multiple computers

Support applications developed in different languages

Support applications running on different types of OS's

Support applications which have GUI and those which do not

(for example Message Interface testing)

Support applications which use different types of network protocols such as TCP/IP, DDS, etc

Support integration of multiple commercial testing tools from different vendors and open source (allowing as new or better products emerge in the market they can be utilized)

Support testing w/o having to install ATRT on the same computers as the application under test and be able to be distributed across computers



The MIT 2009 Information Quality Industry Symposium



Summary

- Apply Hardware Automation efficiencies – Component Reuse; Autonomous computing; self-testable components;
- Borrow from Software Development Advances
- Continued R&D&T is Needed
- We are Continuing to Improve our AT & Continue to Automate the Automation – remove human involvement
- You can read more about this in *“Implementing Automated Software Testing”* Addison Wesley, Mar 2009



The MIT 2009 Information Quality Industry Symposium



Questions...?

Thanks!