

eValid Technology Summary

Automated Testing of Modern Web Applications

Business White Paper

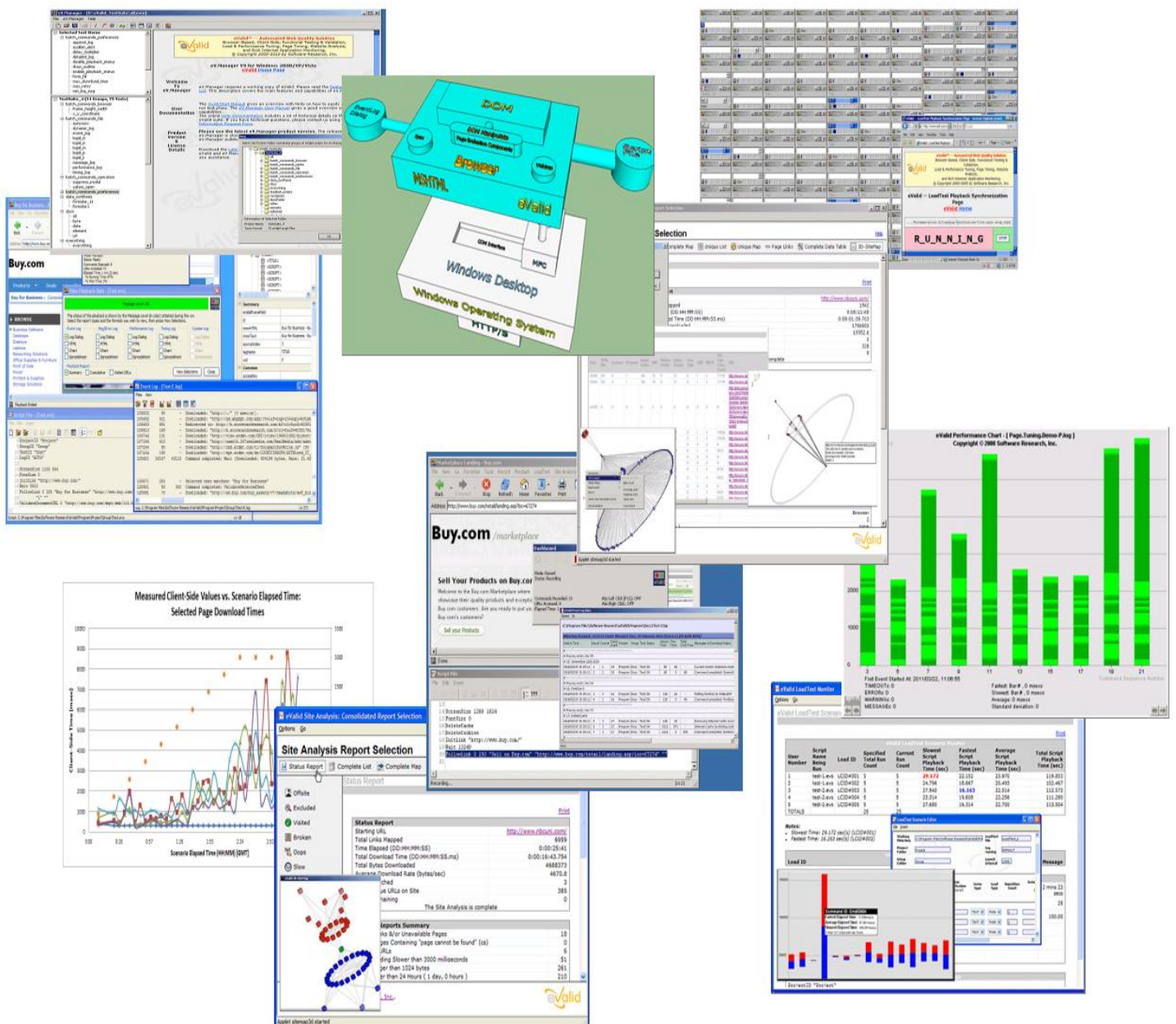


Table of Contents

Executive Summary -----	3
Introduction -----	3
Intrinsic Challenges of AJAX and Web 2.0 -----	4
The Problems with Traditional Testing Tools -----	5
eValid Technology -----	5
Problems in Scripting -----	6
Benefits of eValid -----	7
How eValid Technology Works -----	8
Conclusion -----	10
References -----	11

Executive Summary

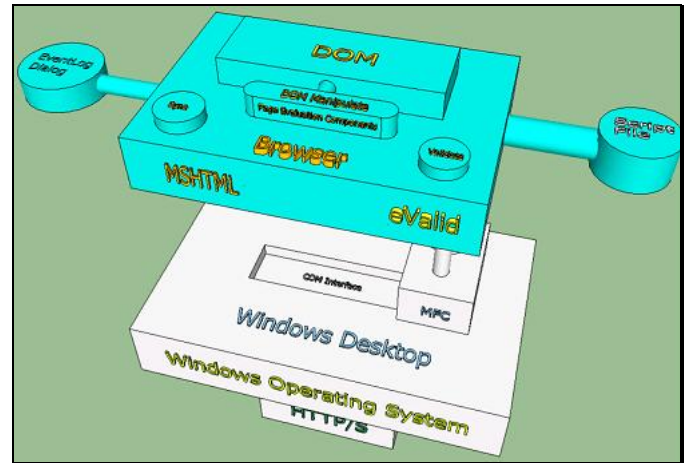
Web applications are developed today with a range of AJAX and Web 2.0 technologies. Powerful new tools and technologies offer advanced features for building user-friendly, highly interactive, desktop and mobile applications that can help companies stay far, far in front of their competition by providing quality end-user experience.

At the same time, these advanced-technology web applications can often pose critical new risks and support challenges. How such Rich Internet Applications (RIAs) operate makes it very important – even critically important – to test them well. That is, to accurately and efficiently confirm performance attributes and the user-centric behavior of such complex applications. But the problem arises because of the intrinsic complexity of these new RIA technologies, combined with the ever-growing and changing number of toolkits that support them. And, add in issues arising from the absence of industry standards.

These factors combine make it quite difficult to find effective solutions for functional, performance, capacity, and quality testing.

Traditional testing tools are unable to keep up with the new levels of complexity of RIAs. eValid is our innovative answer to this difficult problem: a browser-based functional plus performance plus regression plus load-testing solution that supports simple Web applications, as well as modern, JavaScript-based AJAX applications built with any available AJAX toolkit. By extracting GUI-level user activity details, based on reading signals from the browser's internal Document Object Model (DOM), eValid makes scripting much simpler. Using eValid, companies can employ non-technical or novice testers for test creation. This advantage frees up valuable technical resources to focus on other critical tasks such as actual test execution, loadtest planning, and results analysis. Furthermore, eValid's approach makes scripting much faster, and this permits your development team to significantly reduce overall test cycle time.

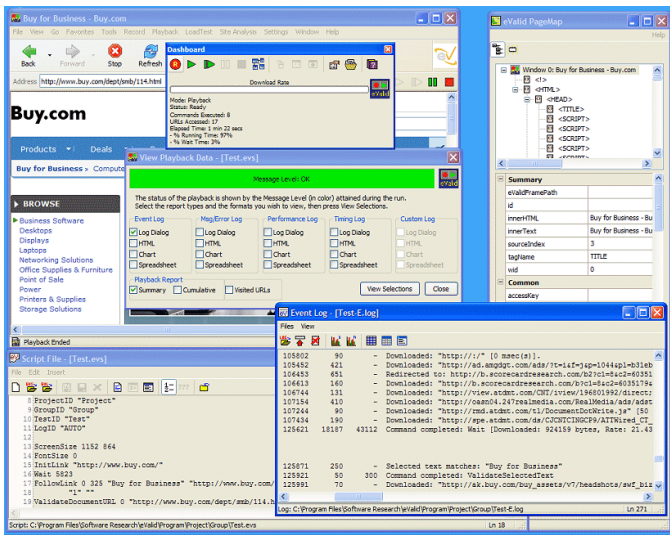
This White Paper explores some of the issues and challenges of quality-testing of AJAX and Web 2.0 applications. We examine reasons why traditional desktop application testing solutions are not a good fit for testing web-based applications that include complex AJAX and other Web 2.0 technologies. We introduce basic eValid technology, and discuss how this revolutionary technical approach helps both to accelerate and simplify web application functional, performance, and load testing for AJAX and Web 2.0 applications.



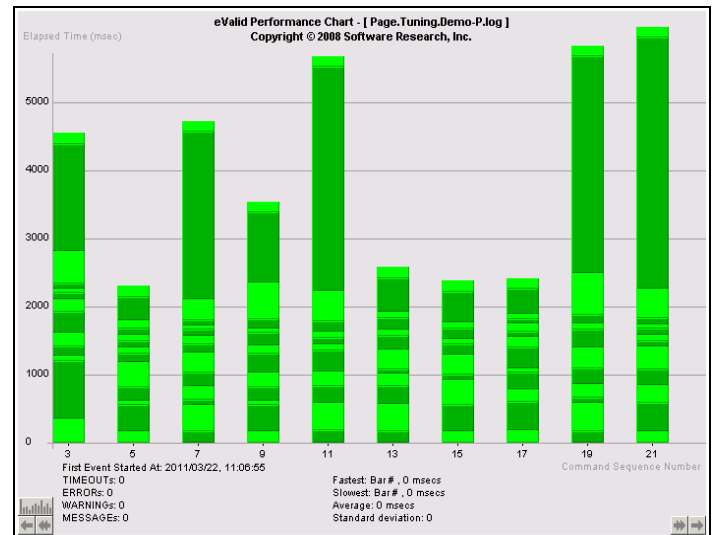
Organization of eValid Product Components

Introduction

AJAX and Web 2.0 applications have recently come to dominate the IT world. They have provided a new way of developing and using Web applications: taking a leap from simplistic, relatively static methods of retrieving and delivering information to highly complex, interactive, user-centered, collaborative Web applications. Many companies are taking advantage of the new techniques and concepts of AJAX and Web 2.0, and are using AJAX and Web 2.0 technologies to help with their application transformation and modernization initiatives. These organizations are building new, modern, interactive front-ends for their core business applications while preserving the underlying business logic and database structure.



Complex Applications Are Analyzed and Tested on The Desktop



Detailed Timing "Stack Charts" For Separate Pages In Website

AJAX -- Asynchronous JavaScript and XML -- and closely related Web 2.0 methods give companies a great way to add a fresh, contemporary look and advanced functionality to their web applications without having to completely rewrite them. Almost all new web applications today are developed using technologies such as AJAX and other Web 2.0 interactive web application development frameworks. The unique advantage of these technologies is that they can transport data "in the background" from the web server without having to reload the entire page. This helps make websites function more like desktop applications and helps create a highly interactive user experience.

As always, with power comes complexity. And innovative technologies can create unforeseen challenges if not properly tested at both functional and performance levels. In Web 2.0 and AJAX applications, the browser clients become "thicker." They are richer in features and functionality, and are more resource intensive. Similarly, powerful features like input auto-complete -- an interactive way of returning a query with every keystroke -- can significantly increase the round trip traffic to the database and greatly impact performance. This can potentially create a considerable number of performance issues, violate application SLAs, and negatively affect the quality of the end-user experience.

Intrinsic Challenges of AJAX and Web 2.0

Rich Internet Applications (RIAs) allow for dynamic, asynchronous data transfer, using multiple protocols and a variety of web application servers. They gather data from distributed, heterogeneous sources, including cloud-based and external data storage options. Thick clients with complex graphic objects and client-side functionality often have server-side components which may need additional processing before the server sends the data back to the client. Developers who build these complex RIA objects -- often adding to them from available toolkits -- do it on their development machines. They often don't realize that once the components are separated across the network, the web server component and page content delivery mechanisms may cause significant delay and thereby may affect overall performance of the application.

New technologies such as AJAX enable features like "prefetching," where (for example) every new letter or clause that a user enters into a search engine suggests a new set of results that are dynamically delivered from the server. All of this activity generates a lot of network traffic and can significantly impact performance. Network latency and bandwidth constraints can also create performance bottlenecks. To accurately predict the performance of an application, it is necessary to test individual components and services, but equally critical are functions like server monitoring and end-to-end performance testing, along with accurate WAN emulation.

Testing Web 2.0 and AJAX applications presents its own set of challenges. The complexity of new technologies, the lack of commonly recognized and accepted standards, and the sheer multitude of emerging frameworks and toolkits make it difficult for companies to build AJAX and Web 2.0 testing strategies and select appropriate automation solutions. Traditional testing tools focus on protocol-level verification, offering no framework-level support or ability to accurately recognize objects in these new, rich clients. This makes it virtually impossible to effectively validate the performance of AJAX and Web 2.0 applications. Script creation, which has traditionally been a lengthy, time-consuming process that requires domain and application expertise, becomes even more complex in AJAX and Web 2.0 applications.

The Problems with Traditional Testing Tools

The emergence of new AJAX and Web 2.0 technologies has also transformed the world of testing. In the very early days, testing tools were based on input/output, protocol-level, recording and playback. That kind of technology recorded the entire HTTP request and response communication between the browser and the connected server. Dynamic values that the server sent back, such as session IDs, had to be inserted mechanically (with great difficulty) if session coherence was to be maintained. As applications became more complex, so did the needed scripting. Manually augmented browser-server state tracking started to require advanced scripting and application expertise, and IT scripting became a complex and time-consuming process -- when it could be made to work reliably at all!

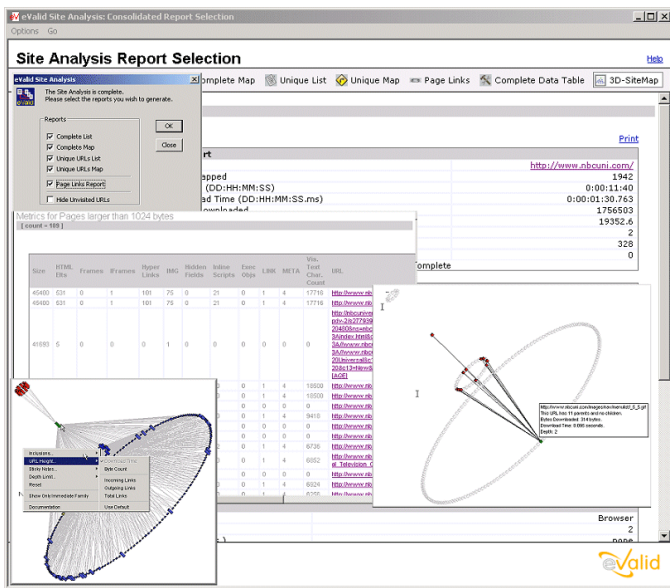
Testing teams then started shifting to GUI-level recording, and ultimately (as in the eValid case) to a focus on verifying specific objects within a browser. Testing tools didn't need to work on the lower transport level layer; they could instead focus on the objects in the DOM. However, AJAX and Web 2.0 have introduced a new set of complexities: client-side processing and asynchronous communication. The GUI-level desktop-based testing tools, even with external state-preservation logic, no longer worked on AJAX and Web 2.0 applications. Some systems included JavaScript enabled "add-ons" to supplement state-keeping.

But that approach in turn presented its own insurmountable challenge: with new toolkits becoming available every month, and old toolkits being constantly updated and revamped, no vendor could keep up and provide a reasonable level of support for the new functionality. Additionally, those conventional GUI automation tools were simply too "heavy" and could automate only a single simulated user per operating system session. A successful functional, regression, performance, and load testing solution would have to have a concurrent multi-user, multi-session driver, automatic state-keeping, simultaneously automating multiple play back capability.

eValid Technology

For the past ten years, Software Research (SR) has been in the forefront of functional, performance, regression, and load testing. With the growing popularity of Web 2.0 and AJAX technologies, SR set out to find a general solution that would support all modern, complex applications, as well as make scripting a faster and easier process. eValid is a completely revolutionary, patented solution that works with the events from the end-user's interactions within the Web application.

eValid supports simple Web (HTML) as well as all kinds of JavaScript-based applications. The eValid scripting engine is fully embedded within the browser, which behaves like a clone of the IE browser. It utilizes a unique, patented approach to object recognition and processing, which makes it an extremely flexible and extensible solution for testing AJAX and Web 2.0 applications.



Site Analysis Function Scans Entire Websites

eValid provides interactive recording and scripting, which makes script creation faster and more trouble-free. It significantly reduces script maintenance effort. GUI-level recording removes the need for programming and correlations, enabling even non-technical business oriented users to participate in the testing process. eValid is completely agnostic to any specific framework or toolkit being used in the application. eValid supports nearly all available AJAX and Web 2.0 toolkits, helping make testing of AJAX and Web 2.0 applications faster, easier, and more comprehensive.

Problems in Scripting

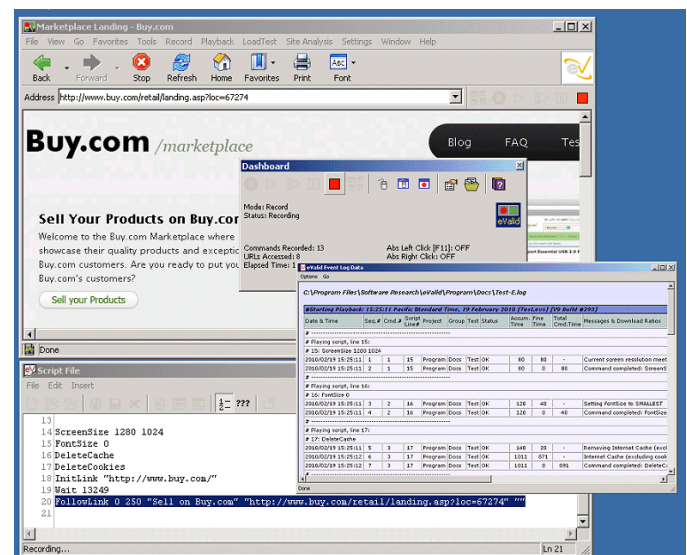
Perhaps the most time-consuming and labor-intensive process in functional, regression, and performance testing is developing test scripts. Although some testing tools have significantly simplified the process of capturing the test script by recording the user's interaction with an application, the fundamental problems with test scripting remain the same:

- Old-fashioned scripting tools require a sequential approach to scripting: first the basic steps are recorded, creating a "shell" script. This script is then taken offline, and undergoes

further manual steps such as data parameterization, followed by additional steps of adding transactions and any other required logic in order to make the scripts perform as desired.

- To create a well-designed, reusable, working load testing script, a tester requires a certain amount of expert programming knowledge. This is especially true for complex, interconnected applications with new technologies that often support critical business transactions.
- Scripts created at the protocol level are vulnerable to the slight application changes. When changes occur to the application, non-adaptable scripts have to be discarded and/or reworked, a huge expense.
- Scripting for AJAX and Web 2.0 applications has to contend with client-side logic and dynamic behavior, asynchronous communication, dynamic DOM modification, and multiple, ever-evolving frameworks and toolkits.

These are difficult challenges that require technological solutions like those embodied in eValid.



Functional Testing Event Log Reports In Real Time

Benefits of eValid

The eValid solution offers significant advantages:

- *Simplified scripting for AJAX & Web 2.0*
Applications: By moving the scripting engine into the browser in an updated and greatly simplified form, eValid technology completely revolutionizes the scripting process. GUI-level recording, based on reading the browser's own DOM, makes it easier for nontechnical users to participate in script creation. The user simply interacts with an application, and the eValid recording engine captures every step in the form of a script. No coding or programming skills are required for capturing and replaying the eValid script.
- *Programming Interface Available: For power users familiar with the C++ programming language, eValid's Programmatic Interface (EPI) offers full programmatic access to work with the script and enhance it on the code level. By combining the ease of use and the breadth of code-level functionality, eValid becomes a versatile tool for both business users and developers alike. eValid, enables novice and nontechnical testers to create load testing scripts, without requiring technical knowledge or in-depth understanding of the scripting process. Now both functional and load-testing scripting can be done faster and easier, saving companies time and freeing technical resources to focus on more advanced tasks.*
- **Accelerated Scripting Shortens Test Cycles:**
eValid technology makes scripting faster, reducing scripting time by as much as 85%, by offering dynamic, interactive functionality that lets users customize scripts during recording. Removing the need to perform multiple sequential steps and time-consuming functions such as state-tracking, and by combining the recording and editing process into one, allows eValid technology to significantly reduce test

cycle time. This innovation is especially welcome in organizations that use Agile methods for delivering applications faster and with higher quality.

By shortening the script creation time, eValid allows more load tests to be performed earlier in the development process, identifying problems earlier in the lifecycle where they are easier and cheaper to fix. eValid scripts are also more robust and intuitive, making them easier to maintain as the application changes. Even more important, scripts used in a functional or regression testing role can be used directly in the eValid load test solution. This capability leads to greater reuse of the testing scripts, increasing the ROI on automated functional and load testing, and improving application reliability and stability. By reducing the amount of time required for test script creation, eValid helps accelerate applications time to market. In today's highly competitive market, these advantages can easily translate into a significant competitive advantage, increased revenues, and improved customer loyalty.

- **eValid supports all AJAX and Web 2.0**
Applications: eValid can precisely and efficiently test all types of AJAX applications. It easily handles a variety of AJAX objects such as slider bars and drop-downs. By looking at the "actual" DOM, eValid automatically inherits compatibility with all supported toolkits as they operate in the eValid browser. Rather than going through a DOM emulator, eValid is based on an event-level record and replay capability, based on direct manipulation of DOM objects.

Instead of recording the state changes to the objects in the DOM, eValid looks at internal event handlers that are causing JavaScript to render in particular ways, and as a result, eValid can achieve a very high level of Web 2.0 and AJAX object recognition. AJAX applications represent business-critical, end-user-facing

functionality. Before eValid, it was practically impossible to effectively and efficiently test these applications. eValid's innovative technology makes AJAX and Web 2.0 testing simpler, faster, and more reliable than ever before.



eValid LoadTest Running 100 Browser Users (BUs) on One Machine

How eValid Technology Works

The recording mechanism in SR's eValid solution is fully interactive and is based on use of an IE-clone browser. When users interact with web applications, they can see the recorded script being developed in real time on the screen. Every user action is sensed automatically and creates playback command sequences that are written in an easy-to-understand English-like script.

Using a unique, patented approach to object recognition, the script automatically recognizes and inserts the names of web objects as they exist in the application. The eValid recording engine also automatically includes data that, at playback time, allows the playback engine to take alternative automation steps via eValid's adaptive playback capability, saving you time on script debugging and troubleshooting.

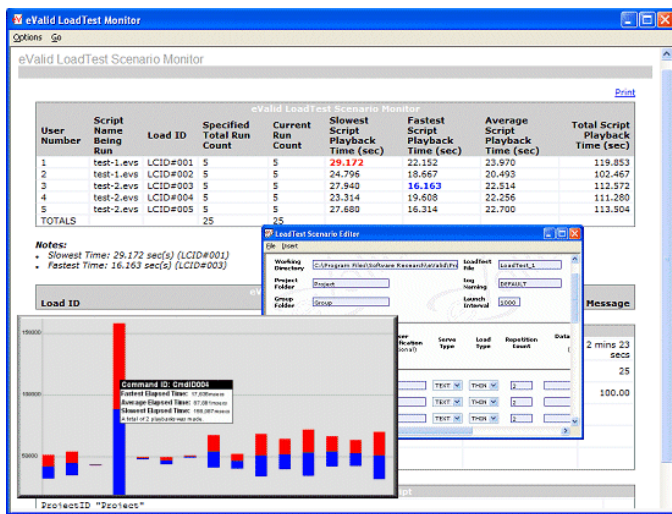
To enable the script to more accurately represent a real-world production environment, you can enhance it with parameters, replacing static data values with variables, and replacing key URLs and other values with environment variable settings.

To capture performance levels, you can add include commands that measures the amount of elapsed time it takes to perform identified functions or sequences of functions. These timer commands can also be recorded from the eValid GUI.

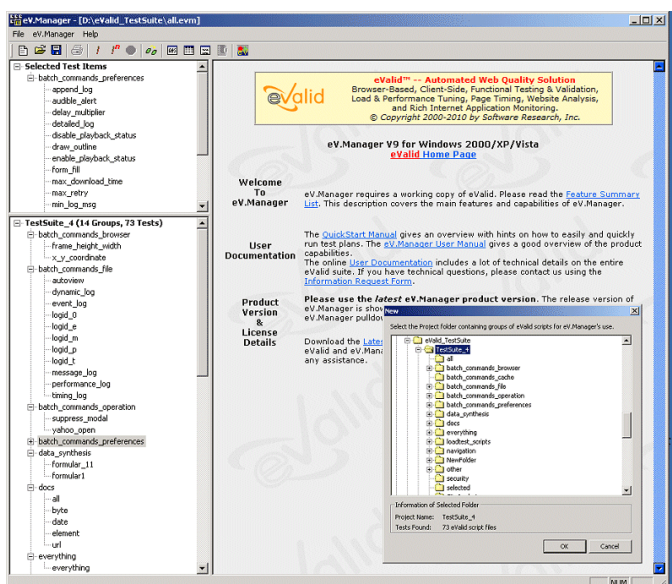
To check the accuracy of the transaction, you can add any number of verification steps. For example, you can simply select a visible text passage and eValid will accept that as the defined value that you need to verify at playback time. Or, you may want to validate that the amount of money that's been added to the account matches the number that you asked to transfer, so you can specify the exact value that you expect at playback time.

To provide greater flexibility in testing a broad range of applications, there is provision in eValid to emulate a range of browser types including those used in mobile applications. This capability allows load testing runs to emulate 1,000's of browser users of any selected range of browser types.

You can add synchronization commands to match the execution of the eValid script with the output from your application. Many essential synchronizations can be recorded directly from the eValid GUI. Synchronization points allow the script to pause in the replay while waiting for an object, a window, a dialog, or another type of control to appear based on a specified condition. By adding special DOM-based synchronization commands, you can accurately capture response times based on asynchronous events such as those which occur in AJAX and Web 2.0 applications. No other technology can do this at this level of detail and accuracy.



Object identification is a major component of the eValid technology. Various functions are available to enable correct object identification, usually based on the use of the eValid PageMap feature to look in detail at the contents of the current web page's DOM. Once you know the index value and properties of an object, you can adjust an eValid script to automatically adapt to a wide range of changes in the structure of the page -- so that your script runs reliably at playback time, independent of unimportant page changes.



Once the script has been recorded, the user can replay it to see if there are errors. eValid has single-step,

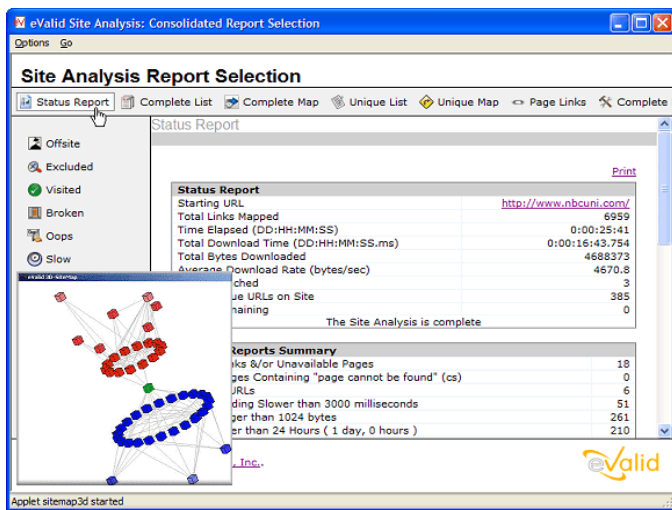
breakpoint, and "insert recording" features that simplify perfecting a script. During test playback, you can see the recorded actions being replayed in the browser. If an error is discovered, the user can highlight the object that failed, and eValid playback logs can help to recognize the step that caused the problem.

Elevation of functional tests into server loading tests is simple and natural with eValid. A built-in load test scenario editor helps compose load test runs, select appropriate run-time parameters, and launch independent eValid instances to act as "Browser Users (BUs)" that impose work on the server stack. Individual machines can run 100's or 1,000's of BUs in parallel. Groups of machines "in the cloud" can simulate 10,000's of BUs.

Once a set of scripts is accumulated for a given application the eValid suite includes a built-in test manager that runs tests on a scheduled basis, in the background or overnight. The user can select a set of tests from the graphical test tree, and can add/delete tests from the work list for a particular test run. Reports from the test manager show PASS/FAIL statistics for all of the tests selected for execution.

An additional capability to analyze complete web sites uses eValid's built-in website spidering function to analyze all of the links in a family of pages, e.g. all those headed by a single URL. Using DOM-based extraction of link information, the site analysis function is very thorough and completely realistic, because the information collected is based entirely on the user's perspective. Data from such runs is presented in tabular reports and in a novel 3D-SiteMap display.

eValid technology combines functional, regression, and server LoadTest modes in a seamless integration, making eValid a true enterprise-level quality and performance testing solution.



Site Analysis Summary Report and 3D-SiteMap Applet

Conclusion

As more and more companies moved toward modernizing their applications and adding new AJAX and Web 2.0 functionality to their core business systems, the product development team at Software Research (SR), beginning in 2001, took on the challenge of completely rethinking how tests are done for web applications, the way events and objects are being recorded, and the way server load experiments are performed. The result is SR's eValid technology -- a truly innovative, browser-based recording and playback engine that supports all kinds of simple Web applications as well as modern AJAX and Web 2.0 websites.

Using a unique and patented, approach to event-level recording and playback for web applications, eValid is able to support all available AJAX and Web 2.0 toolkits. eValid helps make the process of functional testing and load testing of AJAX and Web 2.0 applications simpler, easier, more reliable, more efficient, and more scalable.

eValid makes functional testing and load test scripting faster and easier. It helps shorten the test cycles by integrating the script recording and editing processes.

eValid makes life simpler for everyone involved.

Application testers and performance tuners can customize tests as they record their interactions with an application, removing the need to return to the same

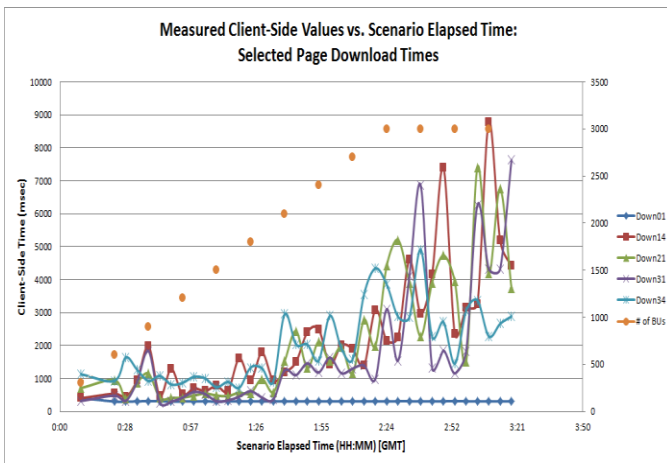
script multiple times to add parameters, data, and other manual fix-ups. This streamlined automated scripting process helps save valuable testing time and shortens the release cycles, enabling organizations to gain a competitive advantage by bringing applications to market faster and with better quality.

Faster, easier, simpler "from the GUI" scripting also makes eValid scripts truly efficient in organizations that use Agile methods. Having faster, more accurate functional, regression, and server loading tests earlier in the lifecycle means fewer problems downstream.

The intuitive, interactive nature of eValid technology and its completely automatic playback synchronization, adaptive playback capability, direct DOM interrogation features, and structural testing approach, enable even novice testers to create working functional and load testing scripts. that handle AJAX and Web 2.0. At the same time, eValid provides full access to "behind-the-scenes" technology and application programmatic interfaces for power users with advanced scripting requirements.

Being able to allocate non-technical resources in test creation and "from life" scripting process translates into significant savings for organizations that are now able to use their highly skilled programmers for more advanced and technically demanding tasks.

Example Loading Experiment



Measured Intra-Script Timings As a Function Of Ramped Server Load

This chart shows the relative download times for five pages within a complex AJAX-based financial analysis web application. (The individual page data is labeled for convenience with the script step numbers 01, 14, 21, 31, and 34) The data was collected during an eValid load test run that ramped the load starting from 1 BU, and increasing the load linearly up to 3,000 BUs over a period of about 2 Hrs.

After the linear ramp-up the server load was kept constant at 3,000 BUs for an additional hour to see if the application performance deteriorated.

The timing data was collected from within each individual BU activity and reported in real time to a central repository. The graph was generated from the individual page data after the run was completed.

Some pages performed much better than other pages. For example, Down34's times varied between 1–5 seconds over the entire experiment interval. Compare this relatively stable performance with the wide variability of Down14, for which the delivery performance deteriorated as the imposed load on the server exceeded about 2,500 BUs.

References

www.e-valid.com

eValid V9 User Manual

US Patent 8,683,447 (25 March 2014)

US Patent 8,650,493 (11 February 2014)

US Patent 8,495,585 (23 July 2013)

US Patent 8,392,890 (05 March 2013)

US Patent 8,327,271 (04 December 2012)

US Patent 7,757,175 (13 July 2010)

US Patent 7,231,606 (12 June 2007)

Additional patents applied for.

(c) 2000–2014 By Software Research, Inc.