Cyberblitz
A bulletin from Computer Science & Engineering
VOL : V
www.rajagiritech.ac.in

FROM HOD'S DESK

BIJU ABRAHAM N, HOD DCS

As we know the effect of recession is coming down slowly. Though the job market has started to come up, the competition is still high. The market has also the people who failed to get placed last year. To take the advantage of the situation, we have to increase our competency and productivity as high as possible through the various opportunities which will help us to improve ourselves. Talking about the increase in productivity we can think of "Aspirations2020" which is a platform for students to prepare themselves to become smart professionals. It will encourage the spirit of competitiveness and accelerate learning through extra- curricular activities, within the student community. This is an extremely important event which will help students to foster problem solving and algorithmic thinking abilities;

"The Great Mind Challenge", which is a Nation-wide Software Development contest conducted by the IBM Academic Initiative Team. This contest focuses on enabling Technical institutes to nurture students. Grooming them to become part of a competitive workforce by arming them with market ready skills.

The CII Innovative Ideas is also another contest where we should take part by follow up action. We have submitted an abstract project, but now we have to submit the detailed project report soon.

Formal Verification of Safety Critical Software

Shimmi Asokan
Asst. Professor

The criticality of software can be classified into three different levels.

- Systems whose failure may cause loss of life or may compromise public safety
- Systems whose failure may cause mission failure
- Systems whose failure may cause inaccurate results or inefficient use of resources

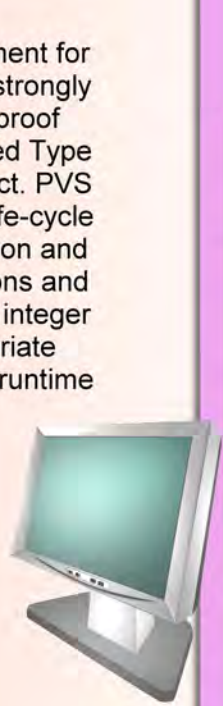
The first type of system is called safety critical system and the second type is called mission critical system.

Reliability of safety critical software is very important. Safety critical software controls the safety functions of the system, so their failures are catastrophic and leads to loss of life. In Safety critical software, failure can have a high price. Such software should be free of errors before it is put into operation. Also the errors should be uncovered in the early stages of development life cycle itself. Traditional verification techniques cannot be relied on as it does not guarantee the absence of errors. One way of ensuring ultra reliable software for safety critical missions is the application of formal methods in the Software Development Life Cycle.

The traditional verification and validation methodology for mission critical software relies on testing, peer review, abstract models and simulations. Engineering judgments and the tradition that has evolved from experience also plays a major role in the V&V of mission critical software. There are two approaches to verify the correctness of a program. The first method is to demonstrate that the program does not meet its specification by finding a counter example. Software testing and inspection processes belong to this category. The effectiveness of these techniques depends on the selection of test cases. Absence of defects within the functional limits can be ensured only through extensive testing with ideal test cases. The effectiveness of these methods is also prone to human errors. Hence these methods are not sufficient for the verification and qualification of safety critical software.

The second method is to use proofs and formally verify that the program is correct with respect to its specification. Formal methods belong to this category. Formal verification techniques use a set of mathematics-based tools and techniques which conduct an exhaustive exploration of all the possible behaviors of the system . Thus formal verification of software provides dependable evidence of freedom from defects. To formally verify a system it should be translated into a formal specification . A formal specification is an expression in some formal language; of the properties some system should satisfy . This specification can be logically evaluated for its correctness. This process of formal verification can be done using any of the available formal method tools. Theorem provers are a class of formal method tools. Theorem provers automate and assist in the process of establishment of correctness of theorems. Many theorem prover tools are available which assists in proving the correctness of a program. Examples of such theorem provers are ACL2, Stanford Temporal Prover (STeP), HOL, Prototype Verification System (PVS) etc. A description of some of the available formal method tools is listed .

PVS, developed by Software Research Institute (SRI) is an integrated environment for developing and analyzing formal specifications. The specification language of PVS is strongly typed and based on higher order logic . Since it supports rigorous typechecking, proof obligations are generated for all type inconsistencies. These proof obligations called Type Correctness Conditions (TCCs) should be proved to prove that the specification is correct. PVS provides formal support for conceptualization and debugging in the early stages of the life-cycle of the design of the hardware and software system. PVS supports both automatic deduction and interactive theorem proving. PVS has a vast prelude library which provides definitions and theorems covering basic mathematical concepts along with the properties of real and integer arithmetic. PVS minimizes the effort of searching the prelude libraries for the appropriate theorems, by making it possible to program proof strategies. PVS can be used to trace runtime errors like overflows/underflows and divide by zero errors of safety critical software.



Media Mining An Emerging Tera-scale Computing Applications

Robin Cyriac
Lecturer

Rapid advances in the hardware technology of media capture, storage, and computation power have contributed to an amazing growth in digital media content. As content generation and dissemination grows, extracting meaningful knowledge from large amounts of multimedia data becomes increasingly important. Media mining is a kind of technology that helps end users search, browse, and manage large amounts of multimedia data. It yields a wide range of emerging applications with various mass-market segments, e.g., image/video retrieval, video summarization, scene understanding, visual surveillance, digital home entertainment, smart health care, etc. Most of these applications are very complicated and have real-time processing demands, which require tera-scale computing power to make them usable.

Media-Mining Usage Model (Personal video retrieval):

A personal video retrieval system is an application that helps end users manage more and more personal multimedia data from all kinds of mobility digital camera devices. In response to a user query, the personal video retrieval application finds the relevant video clips from a large video database. Generally, a retrieval system first extracts low-level audio/visual features from videos, and then detects semantic concepts (keywords) to represent the video content. Finally, a query engine returns retrieval results based on the user's query and on a similarity model. The query can be text keywords, image examples, hand-drawn sketches, or short video clips, and the output is relevant video clips ranked not only by their content similarity to the query, but also by their importance, according to a concept-link relationship analysis. To gradually improve system performance during the query procedure, the system provides user-friendly relevant feedback and active learning modules.

Computational Requirement:

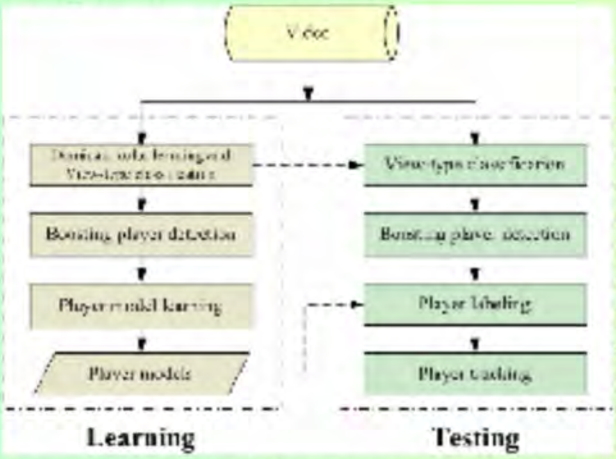


Figure 1: Flowchart of player detection, tracking and classification

In the sports domain, we look at multiple player detection, tracking, and classification in broadcast soccer video for our example. Its flowchart is shown in Figure 1. To make the algorithm robust and adaptive, we construct the background (playfield) color model and three player appearance models (Team A, Team B, and Referee) through unsupervised learning procedures. In the learning phase, the background color model is obtained by accumulating color histograms over hundreds of frames in the video in HSV color space. Player appearance models are learned by player sample collection with a boosted player detector, color histogram representation, and clustering. In the testing phase, we first perform background segmentation, playfield extraction, and view-type classification.

Figure 2 is an example of player tracking results, in which white ellipses and rectangles indicate two teams' players and a black rectangle is the referee.

Figure 2: Player tracking results on soccer video

Player detection is achieved by background elimination and a boosted cascade of Haar features. We only show the detailed detection procedure since this procedure is most compute intensive compared to tracking and classification. The cascade detector with multiple stages has the capability of quickly rejecting the regions and focus on the harder-to-classify windows. The number of features selected in each stage is different depending on the expected performance and sampling criterion. Therefore, increasingly complex classifiers are combined sequentially. This improves both the detection speed and efficiency.

- Input: image frame, background model
- Playfield elimination and view-type classification
- Player detection

For each scale
Scan each point to be detected
For each point
Evaluate its response with cascaded stages
Calculate normalized constant
For each stage
Evaluate the response
For each selected Haar feature
Calculate Haar feature response
Normalize Haar feature response
Get weak classifier response
Accumulate all Haar response
If verified by the threshold, begin next stage;
else, label the point as negative, break;
If pass all stages, label the point as positive

- Post-processing to merge adjacent detection instances
- Output: vector of player regions (rectangles)

Based on the above description, one can easily infer its computation complexity, which is proportional to the size of the video frame, the number of weak classifiers, and the number of scales. For player tracking between two adjacent frames, it is proportional to the number of players and player size. For player classification, it is linear to the number of players, player size, size of codebook, and size of sub-model. For an MPEG-2 video, the frame size is 720x576; we use about 1000 weak classifiers and three different scales. Thus, one minute of MPEG-2 video will need 1.86 tera-operations.

Virtualization

Shaun Thomas
Associate Lecturer

In the last year or two we have seen virtualization go from a poorly understood concept to a much-hyped industry buzzword being bantered about constantly in every conversation involving technology. There is no doubt that virtualization is playing an important role in today's IT landscape, and it even applies to the small and medium business markets. Unlike many technologies that provide a great degree of technological risk and expense and may not be appropriate for a small business, virtualization is a mature technology that is well understood. In short, it provides a layer of hardware abstraction that can benefit an IT organization of any size. It may possibly apply even more to small business IT departments than to the enterprise space.

In today's IT landscape it has become popular to re-label many common technologies as "virtualization" for marketing reasons, unnecessarily complicating the issue. True virtualization refers to the virtualizing of entire operating systems. We can refer to virtualization as platform virtualization. The basic concept of platform virtualization involves running an abstraction layer on a computer that emulates the hardware itself. Through the combination of abstraction and emulation we get what is known as a virtual machine. This virtual machine is a completely working "computer" onto which we can install an operating system just as if we were installing onto the bare metal of a dedicated machine.

Platform virtualization has been available since 1968, why is it only becoming popular and important recently?" The answer is actually quite simple. Traditional platform virtualization technologies require a lot of support within the computer hardware itself. IBM has been building this type of support into its mainframe systems for decades. Large UNIX vendors like Sun have been providing this in their high-end UNIX servers for years as well. These systems are highly specialized and typically run their own custom operating system. Generally only large IT shops could afford servers of this magnitude. Small shops did not have ready access to these technologies. For those IT professionals who have worked with this type of equipment in the past the idea of virtualization was often so ingrained into the platform that it was often discussed very little. It was seen as simply an aspect of these high-end server systems and not necessarily a concept in its own right. What has changed recently is the move to bring platform virtualization to the commodity hardware space occupied by the AMD and Intel (x86_64).

The first move was to use software alone to make this possible on the x86 processor family. The early players in this space were VMWare and Microsoft, with products like VMWare Workstation, Virtual PC, VMWare GSX and MS Virtual Server. These products showed that no special hardware was needed to effectively virtualize whole operating systems. Companies of all sizes began to experiment with the concept of virtualizing their existing commodity platforms. This form of virtualization is known as "host-based virtualization" as it requires a host operating system on which the virtualization environment will run. Following on the tail of these software-only solutions, the big processor vendors in the commodity space, AMD and Intel, began building virtualization capabilities into the processor. This allowed for more flexibility, security and performance. It brought the commodity x64 hardware market much more in line with the traditional offerings from the other processor families common in big iron servers. By doing so, the virtualization market has really exploded. This is true both from the vendor side, as more and more vendors begin offering virtualization related products, and from the customer side, as virtualization begins to be better understood and its use becomes more commonplace. With the latest rounds of purchasing, most IT shops have purchased servers, and often desktops, that support hardware-level virtualization even without intending to prepare themselves for a move to virtualization, making the equation often tip in that direction naturally. This hardware-supported virtualization model is called "hypervisor-based virtualization" as all operating systems run on top of a tiny kernel called the hypervisor and no traditional operating system runs directly on the hardware.

Businesses, even small businesses, will immediately see many advantages from virtualization, even doing so on a small scale. Some of these benefits are obvious and some are less so.

•**Cost**: Our first advantage is hardware cost. By eliminating the need to purchase and support expensive server hardware on a per operating system basis we can now deploy more systems at lower cost per system. In many cases this is not only a cost savings but will also provide greater funds necessary to move from more Spartan servers into fewer – but more enterprise class – offerings with important performance, stability and support features. These features may include -integrated power management and KVM over IP from an out-of-band management console.

•**Reducing power consumption**: It is very trendy, and for good reason, for companies to be concerned with how "green" they are today and IT virtualization plays a key role in the greenification of the department. The addition of virtual machines onto a single physical server typically represents a trivial, if even measurable, increase in power draw. Adding additional physical servers, of course, adds a significant amount of power consumption even for systems that are lightly used or used only occasionally.

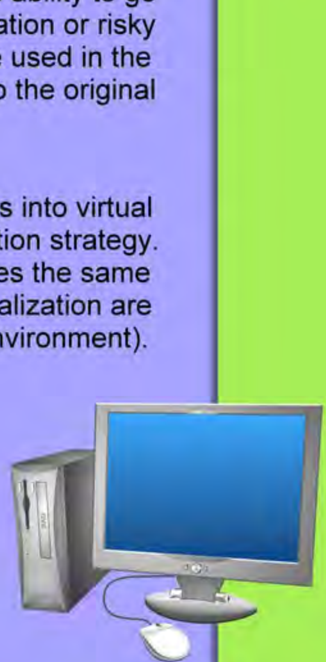
•**Reducing backup complexity**: Virtualized servers can be backed up using completely traditional methods such as file system level backups from the operating system itself as made popular by traditional backup systems like NetBackup, BackupExec, Amanda, Bacula, and others. So if we desire to stick with current backup strategies we can without any additional complexity, but if we want to move to image-based backups we can do so quite easily. Using system images as backups is not necessarily new or unique to virtualization but virtualization makes this far more obvious and accessible for many users. In fact, with virtualization system images (a copy of the entire system, not just of its individual files) can be taken using nothing but the regular filesystem - no special software needed. A complete system backup can be taken by simply shutting down the virtual server, making a copy of its virtual filesystem - often a single, large file, and starting the system up again. Restoring a system can be as simple as copying an image file from a backup storage device to the virtual server.

•**Ease of provisioning**: Building a new server operating system directly on hardware is a time consuming venture for most shops. This is especially true if there are any surprises with new hardware type that has not been used previously. There may be missing drivers or special operating system settings and parameters needed to support the hardware. With virtualization the target platform is always identical, removing many surprises from this process. This makes it both faster and more reliable. In many cases deployment is also faster simply because the process of preparing the base machine is so much faster. To kick off a manual install of Linux on a traditional physical server I must purchase the server, install into rack, connect power and networking, provision networking, turn on server, update firmware, configure out of band management system, burn in hardware, install installation media and begin installing. Or from some virtualization environments I can simply kick off the entire process with a single command at the command line. Deploying a new server could go from hours or days to minutes. This does not even begin to address the simplicity of cloning existing systems within a virtual environment.

•**Significant software cost savings:** Some vendors, like Novell with SUSE Linux, allow you to virtualize as many servers as you want on a single physical machine while paying for only a single machine license. Red Hat gives you multiple installs but not unlimited like Novell. Microsoft has a range of virtualization pricing options depending on your needs, including an unlimited per processor deployment license. In a worst case scenario you will need to pay for additional operating system and other software licenses exactly as if you were running the same machines physically but in almost all cases there is more pricing flexibility and often dramatic cost reductions for multiple virtualized hosts.

•**The ability to "roll back" an entire operating system:** Most virtualization platforms allow for a concept of taking a system snapshot, making changes to the active system and then restoring the system back to its original state when done. This is great for software testing. It's especially good for testing operating system patches or any critical update process where, if something went wrong, it could cause your system to become unresponsive and potentially not repairable. The ability to go "back in time" to the latest snapshot, taken seconds before the patch application or risky configuration change can be a lifesaver. Of course taking an image backup could be used in the same way but snapshots allow for even more rapid recovery due to their "proximity" to the original file system.

All virtualization technologies perform the same function: Abstracting physical resources into virtual ones. This fact makes it more, not less, difficult to choose a particular virtualization strategy. VMware offers a free version (ESXi) of its flagship ESX product that includes the same enterprise-level features but without the operating system. Other choices for server virtualization are Citrix XenServer, Microsoft's Hyper-V, and the little-known ProxMox VE (Virtualization Environment).



Xavier Varricatt
S5 CS B

The Acid Test:
Can your browser do it?

The internet is a battle field for the browser wars that is taking place between Internet Explorer (Often abbreviated as IE), Firefox and Chrome. While major corporations flex their muscles with new and incompatible web features in the battle for the ultimate surfer, are the users of these browsers really suffering from these corporation's attempts to jump away from the global standard?

With each new browser and its iteration, new features and standards that have been exclusive to the browser have been implemented and pushed to users without the recommendation of the World Wide Web Consortium. By implementing these standards and features exclusive to them, these browsers stand to gain an exclusive 'must have' feature which if web developers choose to implement, will force users with no choice but to use their browsers for surfing the internet.

While most users right now do not observe too many issues with our browsers, there are small inconsistencies and changes between the renderings of the same site. IE6 did not have proper support for CSS and hence some sites which were rendered properly in Firefox were rendered rather poorly in IE6 and vice versa. Fonts of the text in IE look much smoother than in Firefox and may even be different. The reason for these differences is that each browser uses its own set of standards and layout engines for rendering a given HTML page, besides the settings that each browser uses for rendering pages by default.

To ensure the proper use of the standards in the internet, a group of web developers called WaSP (Web Standards Project) have tried to encourage the use of the various web standards officially accepted by the World Wide Web Consortium. They have set up various task forces to ensure that these browser developers follow the global standard for page rendering. This group has made a series of web pages known as the Acid tests to verify the implementation of global standards from a web browser.

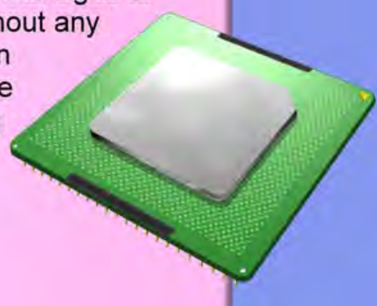
The Acid1 reference image

The Acid1 test originally called the Box Acid Test was made in October 1988 by Todd Fahrner. This was made before WaSP was formed and was made as a comprehensive test to ensure that web browsers comply with the web standards of the time, such as Cascading Style Sheets 1.0 (CSS 1.0). CSS is used to separate document content from document presentation, i.e., rather than defining the font and style of each part of the page every time in the page's code a separate sheet called the Style sheet is used. This allowed web designers to work more on the presentation rather than the fonts of the page. CSS has many applications for web browsing. As the Style sheet can be easily replaced with a user generated one, it helped provide support to Braille, larger fonts for readability, etc allowing people which blindness, short sight, and such disabilities to be able to browse the web. As of now most browsers easily pass the Acid1 Test, implementing CSS 1.0 in their rendering engines. Since 1999, this test is now a part of the CSS1 Test Suite.

The Acid2 reference image

As generated by Firefox 1.0

The Acid1 test paved the way for the Acid2 and Acid3 tests. The Acid2 Test was a compilation of then commonly used standards such as HTML markup, CSS 2.1 styling, PNG images, and data URIs that web developers would've liked to use. Just as in Acid1, a reference image was used to verify if the browser succeeded the test. If a browser manages to implement these features properly a smiley face appears without any distortion with the text "Hello world!" above it. CSS 2.1 was an improved standard over the original CSS which included more styling options such as absolute relative and fixed positioning of elements and much more.



Version 2.1 was also a fix for the original 2.0 standard which had a few issues. PNG or Portable Network Graphics was then used to transfer images over the internet. It was a lossless compression method to store images and did not require licensing unlike the GIF format. Data URIs is a URI scheme with the ability to include data items in-line in a web page as if they were being referenced as external resources. As of now due to security issues of embedding data this way, many browsers such as IE6 and IE7 still avoid implementing this fully.

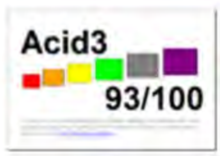
The Acid2 test consists of a text string at the top to ensure proper CSS 1 support. The face appears in a containing block (Not visible to user) and the face itself is comprised of 14 rows, with each row testing various aspects of each technology. The first row tests for minimum and maximum heights and widths as well as for relative positioning of the elements in that row. The next row tests for attribute selectors, class selectors, absolute positioning and floating elements. This row consists of three boxes, two black and one yellow. The third row tests width, overflow, universal selector and data URIs. The next two rows test for paint order and fixed backgrounds. The eyes are PNG images. The next three rows (6-9) check for generated content and child selectors. The nose is actually two black triangles stacked on top of each other, giving the appearance of a diamond. Rows 9 and 10 contain many vertical margins to test vertical margin collapsing. Rows 10 and 11 also test vertical collapsing as well as nested floating elements. Row 12 tests the line height property and Row 13 tests parsing, cascading and selectors. The last row tests for CSS tables. Also, when the mouse hovers over the nose, it is expected to turn blue in color.

While the entire testing process might be too much to explain here and most of you might not even understand half of what was typed in the last paragraph, the systematic and intriguing effort taken to test the various standards that the browser is expected to support does deserve a lot of appreciation. If you know a little about web development, you can read more about this test at http://www.webstandards.org/action/acid2/guide/

When this test was released on April 2005, every browser available at the time could not generate this page properly. Even major browsers at the time such as IE6 and IE7, Firefox 1.0 and 1.5, Konqueror 3.4 could not show the smiley face required and instead showed a distorted page. It was only later in October that year Safari 2.0.2 successfully displayed the page and passed the test. Many browsers followed including Firefox 3.0 and IE8, allowing web developers to create web pages based on these standards.

The Acid3 reference image

As generated by Internet explorer 8.0



As Generated by Firefox 3.5.3

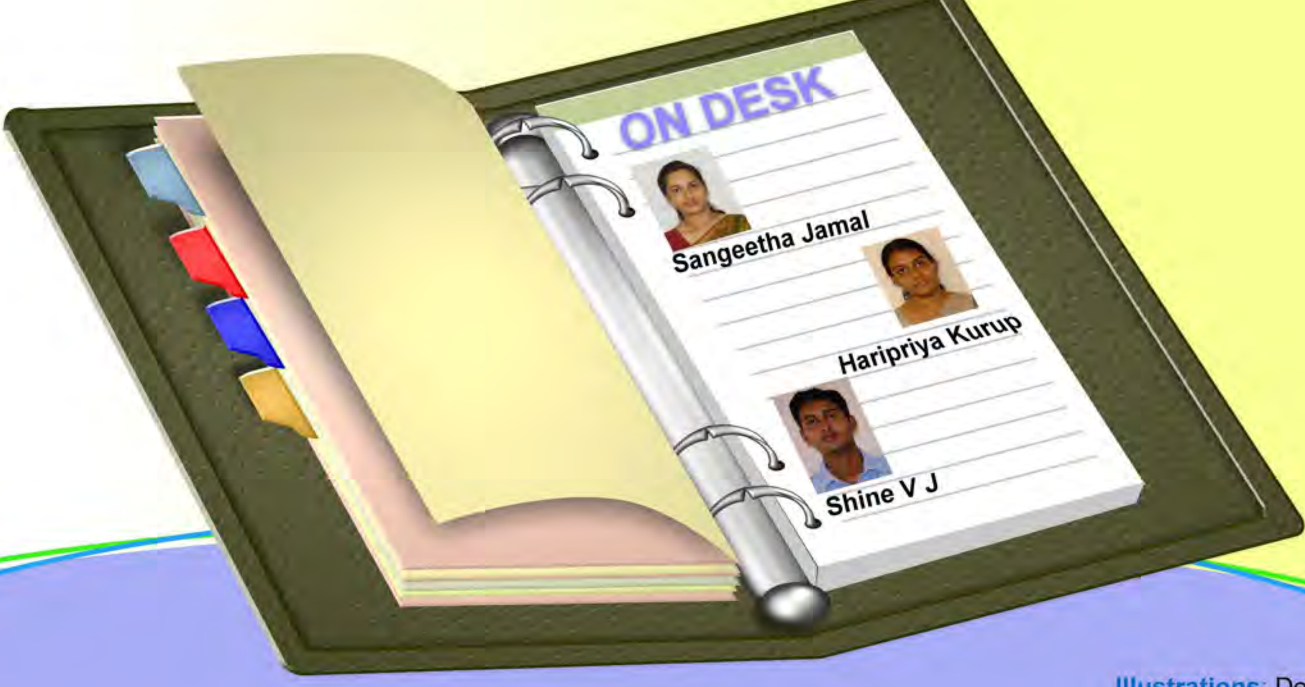
Acid3 test was released on March 2008 and as of now only two browsers have successfully passed the test – Safari 4.0 and Opera 10.00. The Acid3 test has been used to check the compliance of ECMA script, DOM Level 2, Scalable Vector Graphics and XML, as well as Data URIs, CSS3 and HTML. The Acid 3 test was divided into 6 buckets (and hence the 6 bars in the page), each bucket focusing on a specific technology standard. The six buckets are:

Bucket 1: DOM traversal, DOM range, HTTP
Bucket 2: DOM2 core and DOM2 events
Bucket 3: DOM2 views, DOM2 styles, CSS3 selectors and Media Queries
Bucket 4: HTML table behavior when manipulated by DOM and scripts.
Bucket 5: SVG, HTML, Unicode, etc.
Bucket 6: ECMA script

Each bucket consists of a number of subtests for the various features of each technology and based on the number of subtests passed, the corresponding rectangle will change its color from Black to Grey to Silver every 5 subtests it passes and finally to its final color as seen in the reference image once it completes all the subtests in a given bucket.

DOM or Document Object Model is a language independent convention for representation and interaction with various elements in a web page. DOM is used with Java Script to get the status of a web page and modify its contents dynamically. It is useful for detecting user generated events as the user interacts with the web page. Scalable Vector Graphics (SVG) is used for drawing images using an XML file. Unlike most other types of image formats, SVG can be easily resized to any screen size without any distortion of the image. Usually fonts are implemented using vector graphics for easier resizing. XML (Extensible Markup Language) is used to encode documents electronically. It is used in a variety of applications from data interchange over the internet to various documents formats such as Open Office's Open Document and Microsoft Office's Office Open XML. ECMA script is a standardized scripting language made by ECMA international. As of now many companies make their own version of this ECMA script, trying to maintain compatibility with the original script, while adding extra features. These scripting languages are used to add an interactive touch to websites such as managing forms, pop ups, etc. Some versions of ECMA script are Jscript by Microsoft and Java Script by Netscape. A browser which conforms to the various standards tested by Acid3 should be able to generate the required reference image shown and get a score of 100 in its default settings. The browser is also expected to show a generic favicon on the browser toolbar to show proper handling of the 404 error code despite sending an image for the favicon. If a browser fails in some of the subtests of a bucket the letter A in 'Acid3' can be clicked to view the tests or errors that have occurred when generating the page.

There might be an argument at this point that most of our browsers render and generate most of the sites we visit just fine, and that there were only small inconsistencies such as the fonts when viewed by different browsers. It must be noted beforehand that this viewing experience was not the result of the browsers doing a good job of handling the web page, but the needless extra toil and the sweat of many web developers who had to work for hours on end to get the same page to render exactly the same in each of these browsers which come with a unique rendering engine and techniques for interpreting each standard. By ensuring that all theses browsers follow the same set of standards when rendering a page, web developers can rest easy knowing that the same page will render exactly the same, be it Firefox or IE or even Safari. This will also ensure that websites can be developed with interactivity for the user and not incompatibility of a single browser in mind, which in turn will allow for us the end to have a faster, perfect and enjoyable experience over the internet with the latest standards becoming accepted almost immediately by all the major browsers.



Illustrations: Deepu M