

# Something of a Potemkin Village? Acid2 & Mozilla’s efforts to comply with HTML4

Matthijs den Besten<sup>1</sup> and Jean-Michel Dalle<sup>2</sup>

<sup>1</sup> Chaire Innovation et Régulation, Ecole Polytechnique, Paris, France  
`matthijs.den-besten@polytechnique.edu`

<sup>2</sup> Université Pierre et Marie Curie, Paris, France `jean-michel.dalle@upmc.fr`

*The real point here is that the Acid3 test isn’t a broad-spectrum standards-support test. It’s a showpiece, and something of a Potemkin village at that. Which is a shame, because what’s really needed right now is exhaustive test suites for specifications— XHTML, CSS, DOM, SVG.[2]*

Acid3 is the third of three benchmark tests that have been devised to challenge browsers to comply with Internet standards [6]. While Firefox developers at Mozilla had fully embraced the predecessor to Acid3, Acid2, they showed themselves much more reticent this time around. As the quote above indicates they had come to feel that Acid3 would divert attention from the real issues and might actually make it more difficult to achieve “deep compliance” as developers would scramble to come up with quick fixes just to pass the benchmark test. But were these fears justified? To find out, we retrieved the bug reports for bugs in Mozilla’s Bugzilla bug tracker concerning compliance with the HTML4 standard and tried to analyze the differences in the process of bug resolution between bugs that were linked to Acid2 and bugs that were not. In Bugzilla, the bug resolution process passes a number of well-defined stages. Based on the transition rates that we observe we conclude that the process of bug resolution is markedly different for bugs associated with Acid2. In particular, bug resolution appears to be much more chaotic in case of Acid2. This might be symptomatic for “scrambling”, which would explain why developers were not so keen to repeat the experience when Acid3 came around. Further investigations, however, are needed to corroborate this hypothesis.

Bugs reports in Bugzilla are often part of Bug Report Networks [3]. That is, they are part of a network of dependencies as bugs can be declared to depend on, block, or duplicate other bugs. Note that the dependencies between bugs are not always purely technical. In fact, an important type of bugs in Bugzilla is the “meta-bug”, also known as the “tracker bug”, which is a bug at the root of a dependency tree whose leafs are bugs that are related to the issue that the meta-bug is trying to address. For instance, meta-bug 7954 is the bug that tracks issues related to the implementation of the HTML4 standard and the meta-bug 289480 tracks the issues related to Acid2. For our investigation we took advantage of the efforts of the administrators of these meta-bugs to list all bugs related to HTML4 and Acid2 respectively. Yet, as these meta-bugs facilitate coordination among a group of people with a particular interest, much

like project-pages in case of Wikipedia [5], it might be that the differences in bug resolution behavior, which we observe, are a reflection of internal project-culture rather than the effect of external pressure from a public challenge.

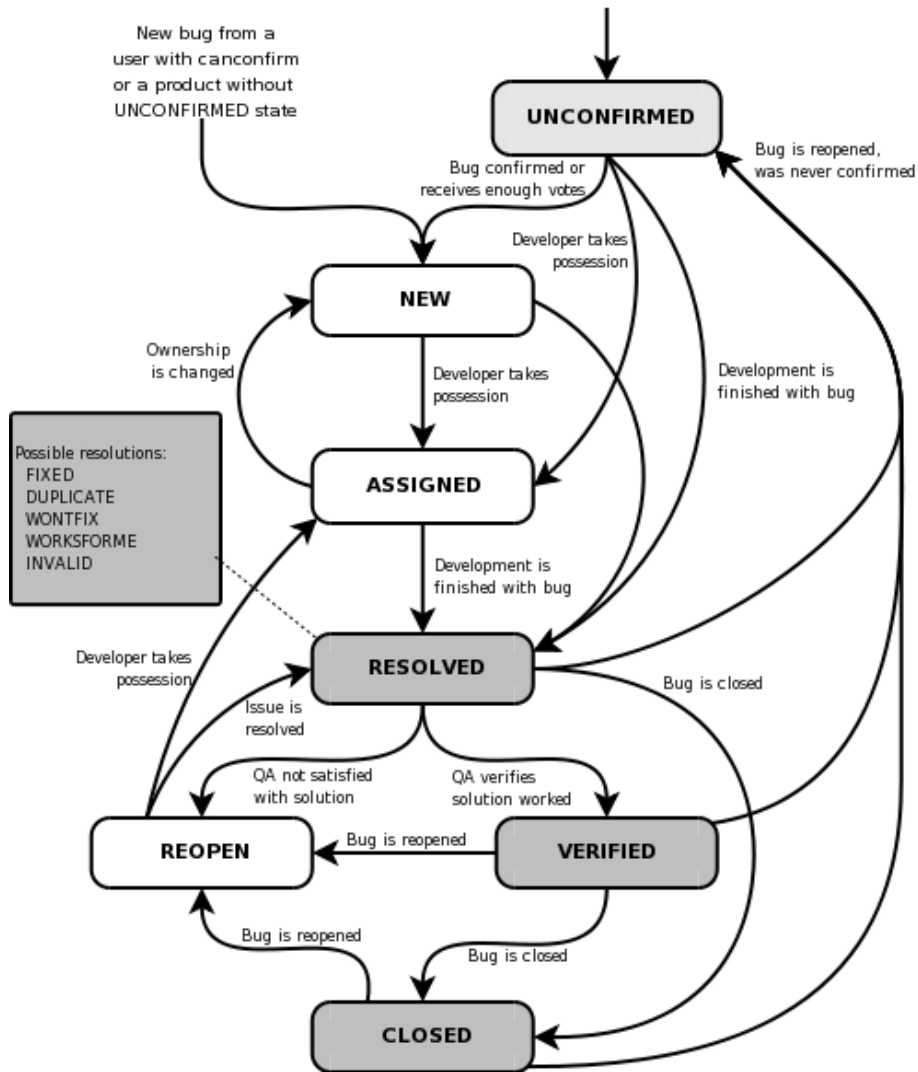


Fig. 1. Bug resolution process according to Bugzilla (source: [4]).

As of March 2011, there are 2904 bugs in the dependency tree of the HTML4 meta-bug and 2195 in the tree of Acid2 (and except for 51 bugs, these bugs also appear in the HTML4 tree). In order to inspect the process of bug resolution, we code each bug report as a sequence of states (cf. [1]), where the duration of

states is defined by the number of messages posted on the bug’s discussion forum and a state by the bug’s Bugzilla bug status. Bugzilla distinguishes 7 forms of “status” [4]: UNCONFIRMED is the default initial state assigned to a bug when it is declared; NEW is the state that the bug gets once it has been confirmed by someone with CanConfirm rights (these people also have the right to declare bugs with an immediate initial status of NEW); ASSIGNED is the status of the bug once it has been “assigned” to someone, making this person responsible for managing the bug resolution process; RESOLVED is the status of the bug once a solution for the problem that it identified has been proposed; VERIFIED is the status of the bug once the solution has gone through a review; REOPENED is the status of the bug signalling that it has been decided that the proposed solution is not valid or not sufficient; and finally a bug can have status CLOSED to indicate that comments to the bug are no longer welcome.<sup>1</sup> There is a canonical path of bug treatment from UNCONFIRMED to NEW, from NEW to ASSIGNED, etcetera, ending with VERIFIED and/or CLOSED that is proposed in the Bugzilla manual (see Figure 1). Deviations from this path are allowed, but that are supposed to be exceptions rather than the rule.

Figure 2 is a state diagram based on the transition rates for bugs related to HTML4, but not to Acid2, on which there was activity between the releases of Firefox 1.5 (November 29, 2005) and 2.0 (October 24, 2006).<sup>2</sup> The shape of the edges indicates the likelihood of a transition.<sup>3</sup> The pathways depicted in the diagram are very close to the canonical path proposed in the Bugzilla manual. In contrast, the state diagram in Figure 3 for bugs related to Acid2 during the same period, which falls just after the launch of Acid2 and includes the passing of the test by a development version of Firefox 3.0 a year later, shows a very different picture full of loops and shortcuts. Further investigations will help us determine whether this is a sign of chaos or due to greater efficacy in solving bugs since the availability of a public test suite makes it easy to verify the resolution of a bug and since people with shared interest come to know each others’ competences.

## References

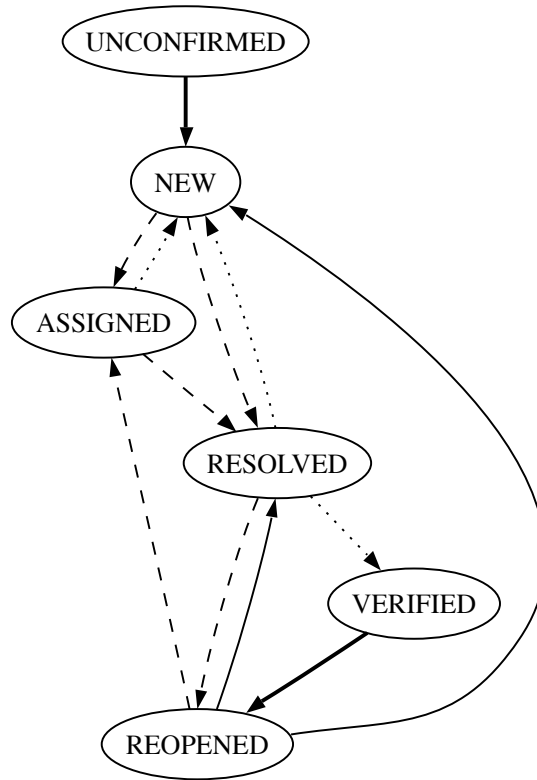
1. Gabadinho, A., Ritschard, G., Müller, N., Studer, M.: Analyzing and visualizing state sequences in R with TraMineR. *Journal of Statistical Software* 40(4), 1–37 (2011)
2. Meyer, E.: Eric’s archived thoughts: Acid redux. <http://meyerweb.com/eric/thoughts/2008/03/27/acid-redux/> (2008), <http://meyerweb.com/eric/thoughts/2008/03/27/acid-redux/>
3. Sandusky, R.J., Gasser, L., Ripoche, G.: Bug report networks. In: *Proc. ICSE Workshop Mining Software Repositories* (2004)

<sup>1</sup> Note, however, that status CLOSED is no longer used (see bug 169885).

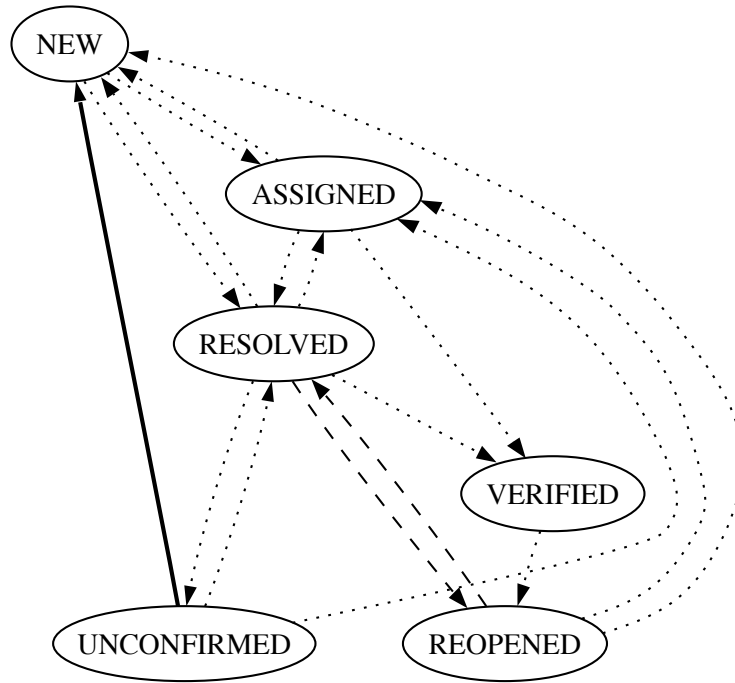
<sup>2</sup> Release dates according to [7]

<sup>3</sup> *Bold* for a rate higher than 0.2; *solid* if  $> 0.1$ ; *dashed*  $> 0.05$ ; and *dotted*  $> 0$ .

4. The Bugzilla Team: The Bugzilla Guide - 3.0.5 Release
5. Ung, H., Dalle, J.: Project management in the wikipedia community. In: Proceedings of the 6th International Symposium on Wikis and Open Collaboration. pp. 13:1–13:4. WikiSym '10, ACM, New York, NY, USA (2010), ACM ID: 1832790
6. Wikipedia: Acid3. <http://en.wikipedia.org/wiki/Acid3> (2011), <http://en.wikipedia.org/wiki/Acid3>
7. Wikipedia: History of Firefox. [http://en.wikipedia.org/wiki/History\\_of\\_Firefox](http://en.wikipedia.org/wiki/History_of_Firefox) (2011), [http://en.wikipedia.org/wiki/History\\_of\\_Firefox](http://en.wikipedia.org/wiki/History_of_Firefox)



**Fig. 2.** Diagram of observed state transitions for bugs related to HTML4, active between November 29, 2005 and October 24, 2006 and *not* associated with Acid2 ( $n = 235$ ).



**Fig. 3.** Diagram of observed state transitions for bugs related to HTML4, active between November 29, 2005 and October 24, 2006 and associated with Acid2 ( $n = 274$ ).