AutoWikiBrowser: Dealing with data sets

Marios Magioladitis

14.05.2017

AutoWikiBrowser: Dealing with data sets

Marios Magioladitis

14.05.2017

Wikipedia: 5,500,000 articles

- **Big Data**: Manual data processing is inadequate to deal with that amount of data
- Great need to apply same rules in all pages
- Many editors, not all familiar with Mediawiki and wikisyntax
- Vandalism
- Typos
- Mass re-categorisations etc.

Mediawiki

- Free and open-source wiki software.
- Originally developed by Magnus Manske and improved by Lee Daniel Crocker.
- It runs on many websites, including Wikipedia, Wiktionary and Wikimedia Commons.
- It is written in the PHP programming language and stores the contents into a database.
- The software is optimized to efficiently handle large projects, which can have terabytes of content and hundreds of thousands of hits per second.

Mediawiki

- Achieving scalability through multiple layers of caching and database replication has been a major concern for developers.
- On Wikipedia more than 1000 automated and semi-automated bots and other tools have been developed to assist in editing.

AutoWikiBrowser

- Created in 2006
- Open-source
- Semi-automated MediaWiki editor for **Windows**
- > 100 code syntax fixes with the use of regular expressions
- Fully compatible with all Wikipedia and thei sister projects
- Allows plugins
- Written in C#

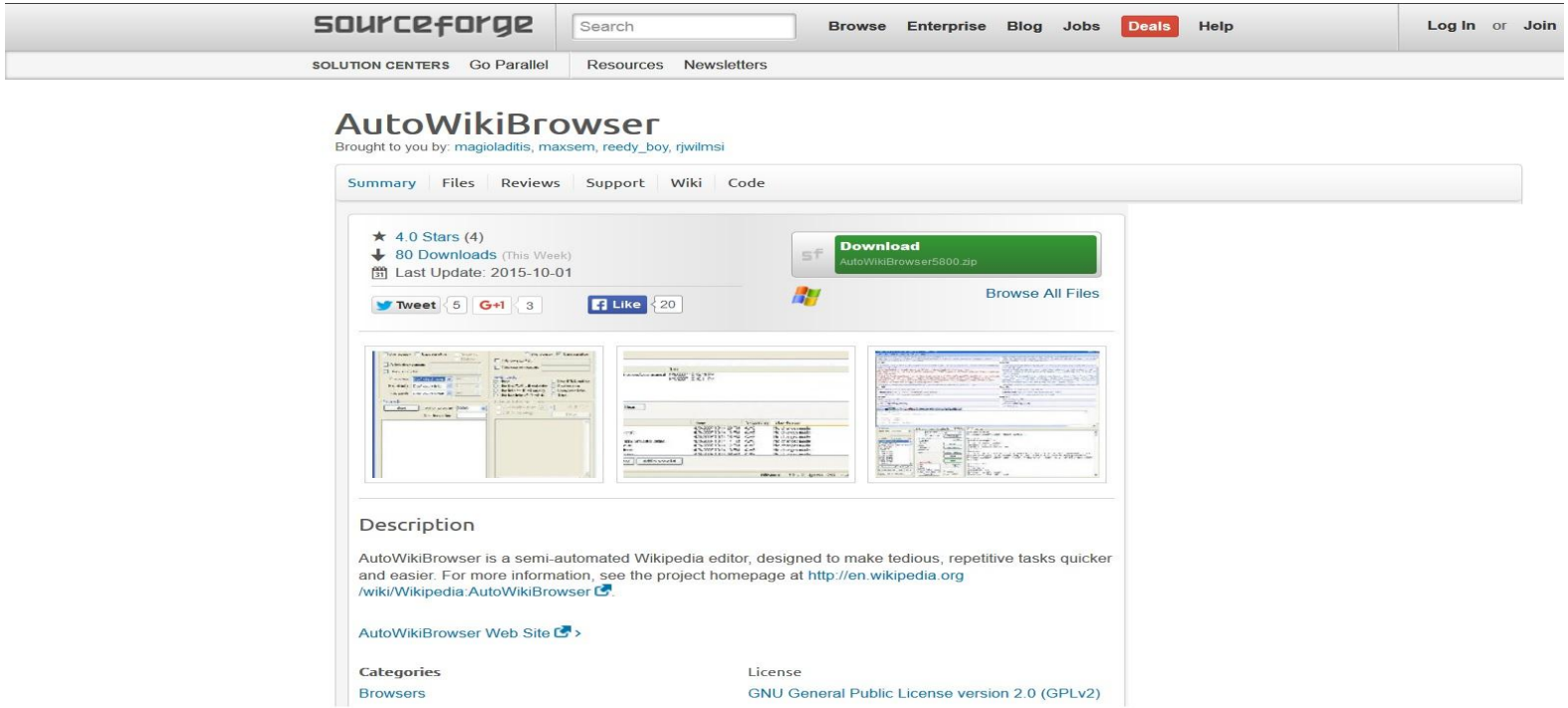
AWB is powerful

- Page edits since English Wikipedia was set up (2001): 852,605,210 (Special:Statistics)
- AWB edits in en.wiki since 2009: 108,633,087 (https://tools.wmflabs.org/awb/stats/)
- AWB edits in en.wiki 2008-2009: 4,904,034 (https://tools.wmflabs.org/awb/stats/)
- Vi.wiki edits with AWB: 193,854,162
- Lietuval.it edits with AWB: 875,640,318

The code at a glance

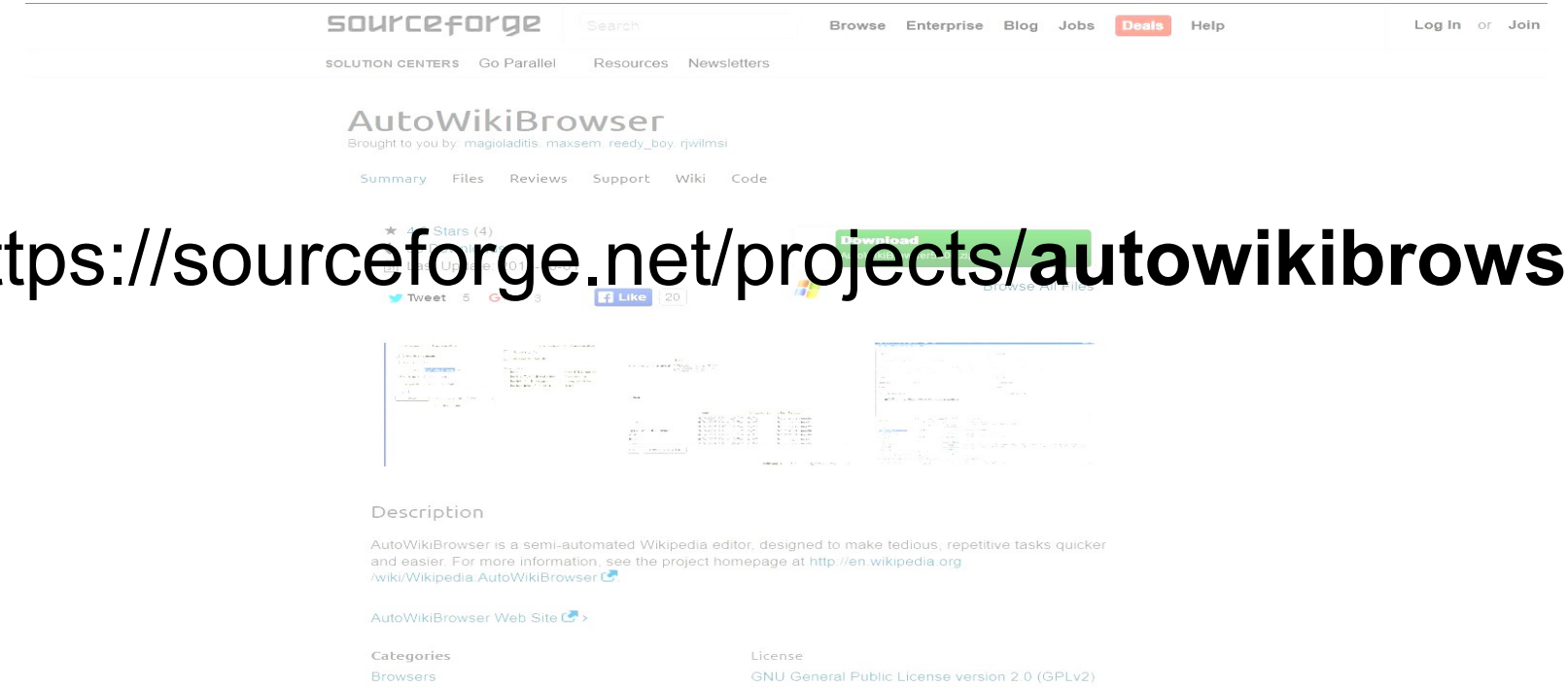
- > 3 millions lines of code [including libraries]
- 633K comments
- 1.6Mb code in C#
- 957 years of effort (COCOMO model)
- Many plugins
- Edit box supports the Microsoft Text Services Framework for use with speech recognition/handwriting applications
- Licence: GPL-2.0+
- https://www.openhub.net/p/AutoWikiBrowser

Code is available at Sourceforge

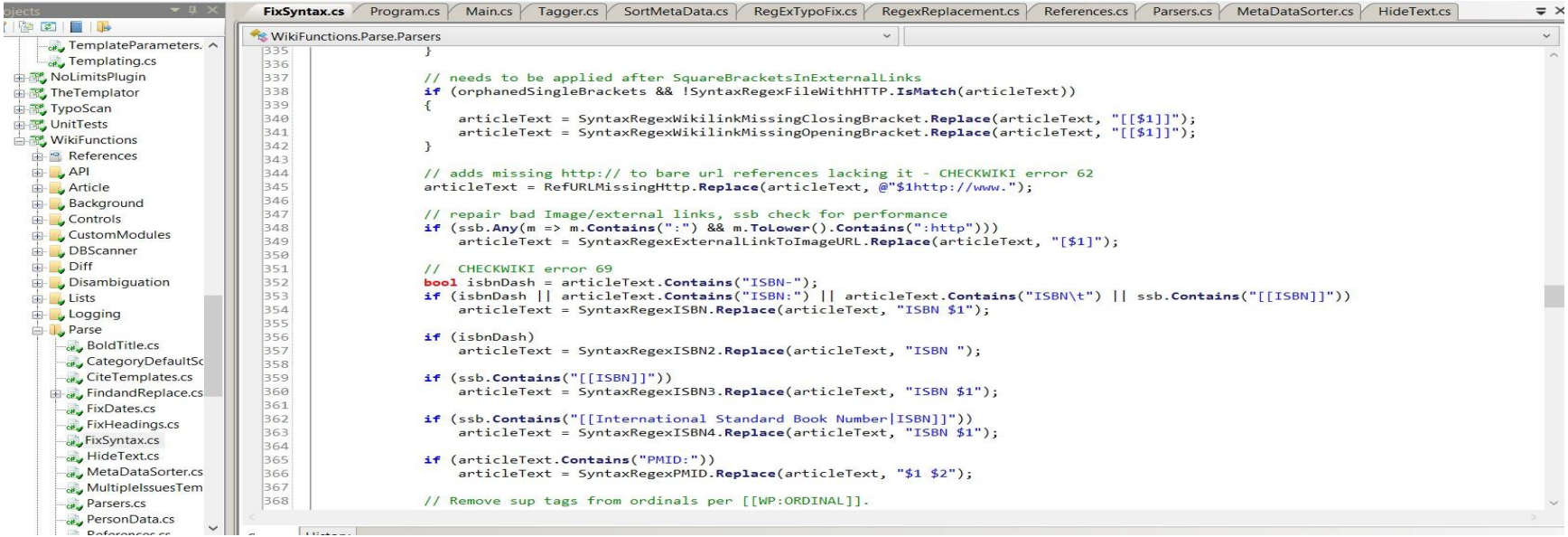


Code is available at Sourceforge

https://sourceforge.net/projects/**autowikibrowser/**



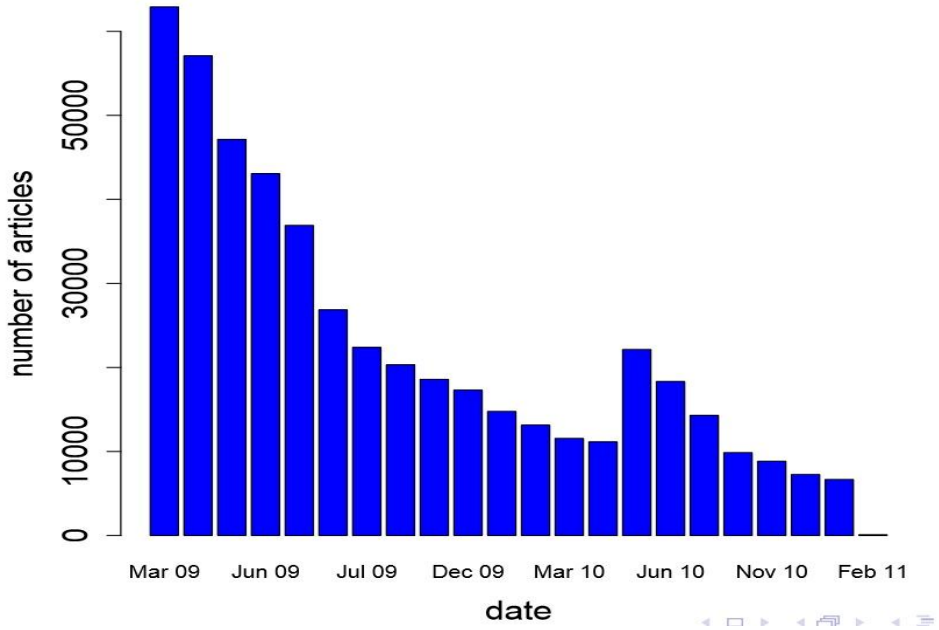
The code


FixSyntax.cs
Program.cs
Main.cs
Tagger.cs
SortMetaData.cs
RegExTypoFix.cs
RegexReplacement.cs
References.cs
Parsers.cs
MetaDataSorter.cs
HideText.cs
WikiFunctions.Parse.Parsers
TemplateParameters.
Templating.cs
NoLimitsPlugin
TheTemplator
TypoScan
UnitTests
WikiFunctions
References
API
Article
Background
Controls
CustomModules
DBScanner
Diff
Disambiguation
Lists
Logging
Parse
BoldTitle.cs
CategoryDefaultSc
CiteTemplates.cs
FindandReplace.cs
FixDates.cs
FixHeadings.cs
FixSyntax.cs
HideText.cs
MetaDataSorter.cs
MultipleIssuesTem
Parsers.cs
PersonData.cs
335 }
336
337 // needs to be applied after SquareBracketsInExternalLinks
338 if (orphanedSingleBrackets && !SyntaxRegexFileWithHTTP.IsMatch(articleText))
339 {
340 articleText = SyntaxRegexWikilinkMissingClosingBracket.Replace(articleText, "[[$1]]");
341 articleText = SyntaxRegexWikilinkMissingOpeningBracket.Replace(articleText, "[[$1]]");
342 }
343
344 // adds missing http:// to bare url references lacking it - CHECKWIKI error 62
345 articleText = RefURLMissingHttp.Replace(articleText, @"$1http://www.");
346
347 // repair bad Image/external links, ssb check for performance
348 if (ssb.Any(m => m.Contains(":") && m.ToLower().Contains(":http")))
349 articleText = SyntaxRegexExternalLinkToImageURL.Replace(articleText, "[$1]");
350
351 // CHECKWIKI error 69
352 bool isbnDash = articleText.Contains("ISBN-");
353 if (isbnDash || articleText.Contains("ISBN:") || articleText.Contains("ISBN\t") || ssb.Contains("[[ISBN]]"))
354 articleText = SyntaxRegexISBN.Replace(articleText, "ISBN $1");
355
356 if (isbnDash)
357 articleText = SyntaxRegexISBN2.Replace(articleText, "ISBN ");
358
359 if (ssb.Contains("[[ISBN]]"))
360 articleText = SyntaxRegexISBN3.Replace(articleText, "ISBN $1");
361
362 if (ssb.Contains("[[International Standard Book Number|ISBN]]"))
363 articleText = SyntaxRegexISBN4.Replace(articleText, "ISBN $1");
364
365 if (articleText.Contains("PMID:"))
366 articleText = SyntaxRegexPMID.Replace(articleText, "$1 $2");
367
368 // Remove sup tags from ordinals per [[WP:ORDINAL]].

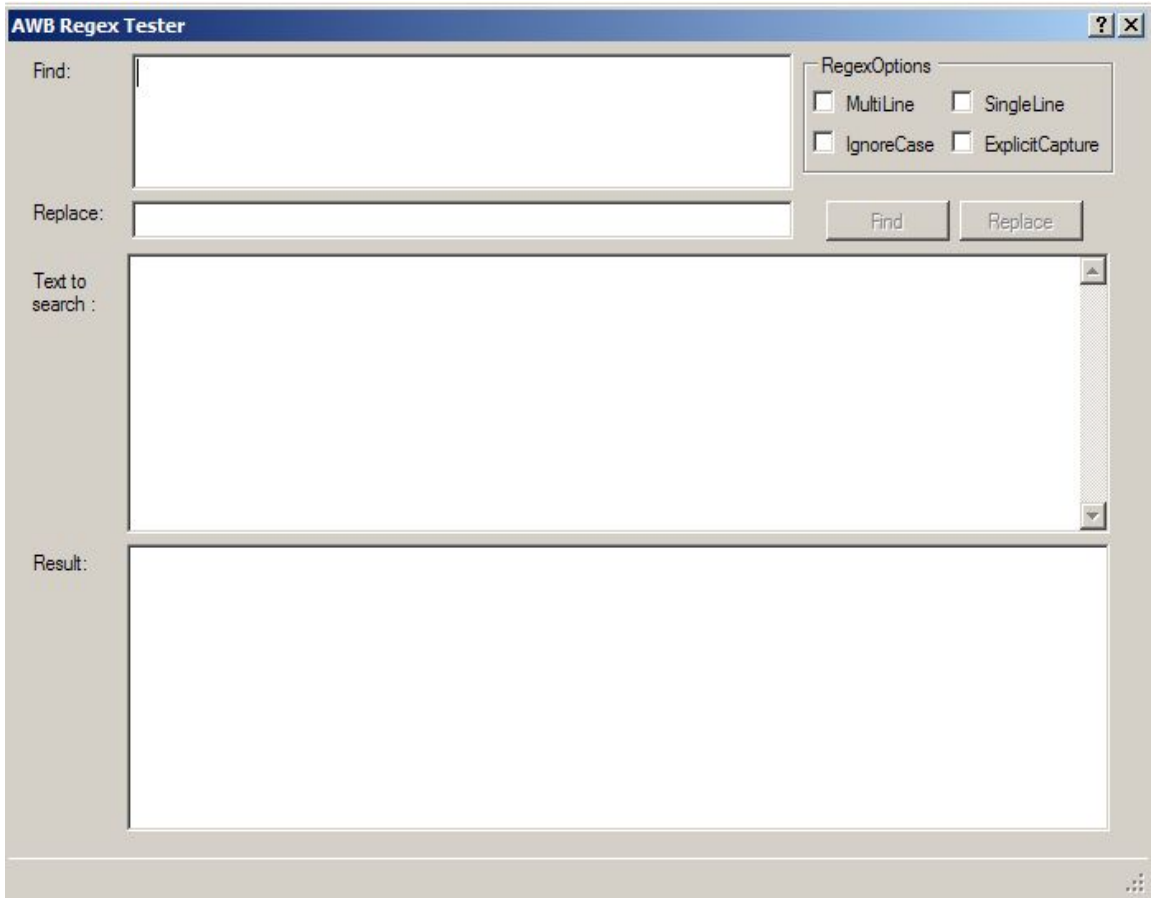
The code


WikiFunctions.Parse.Parsers
FixSyntax(string articleText)
bool badHttpLinks = Tools.DeduplicateList((from Match m in HttpLinks.Matches(articleText.ToLower()) select m.Value).ToList()).Any(
if (badHttpLinks)
articleText = MultipleHttpInLink.Replace(articleText, "$1");
articleText = MultipleFtpInLink.Replace(articleText, "$1");
if (badHttpLinks && TemplateExists(alltemplates, WikiRegexes.UrlTemplate))
articleText = WikiRegexes.UrlTemplate.Replace(articleText, m => m.Value.Replace("http://http://", "http://"));
if (badHttpLinks && !SyntaxRegexHTTPNumber.IsMatch(articleText))
{
articleText = MissingColonInHttpLink.Replace(articleText, "$1://$2");
articleText = SingleTripleSlashInHttpLink.Replace(articleText, "$1://$2");
articleText = articleText.Replace("https://http://", "https://");
}
if (CellpaddingTypoQuick.IsMatch(articleText))
articleText = CellpaddingTypo.Replace(articleText, "$1cellpadding");
if (SimpleTagsList.Any(s => s.Contains("font")))
articleText = RemoveNoPropertyFontTags.Replace(articleText, "$1");

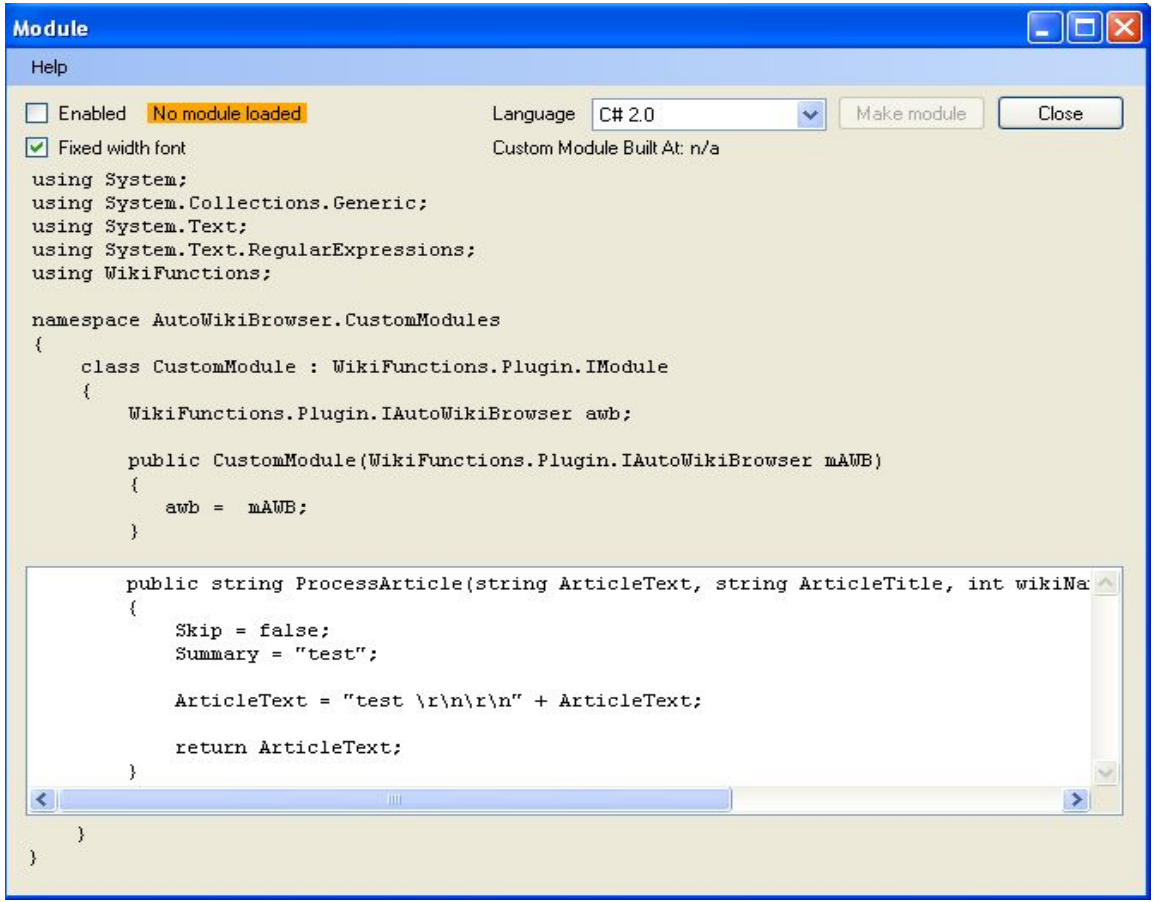
An example: BLP tags



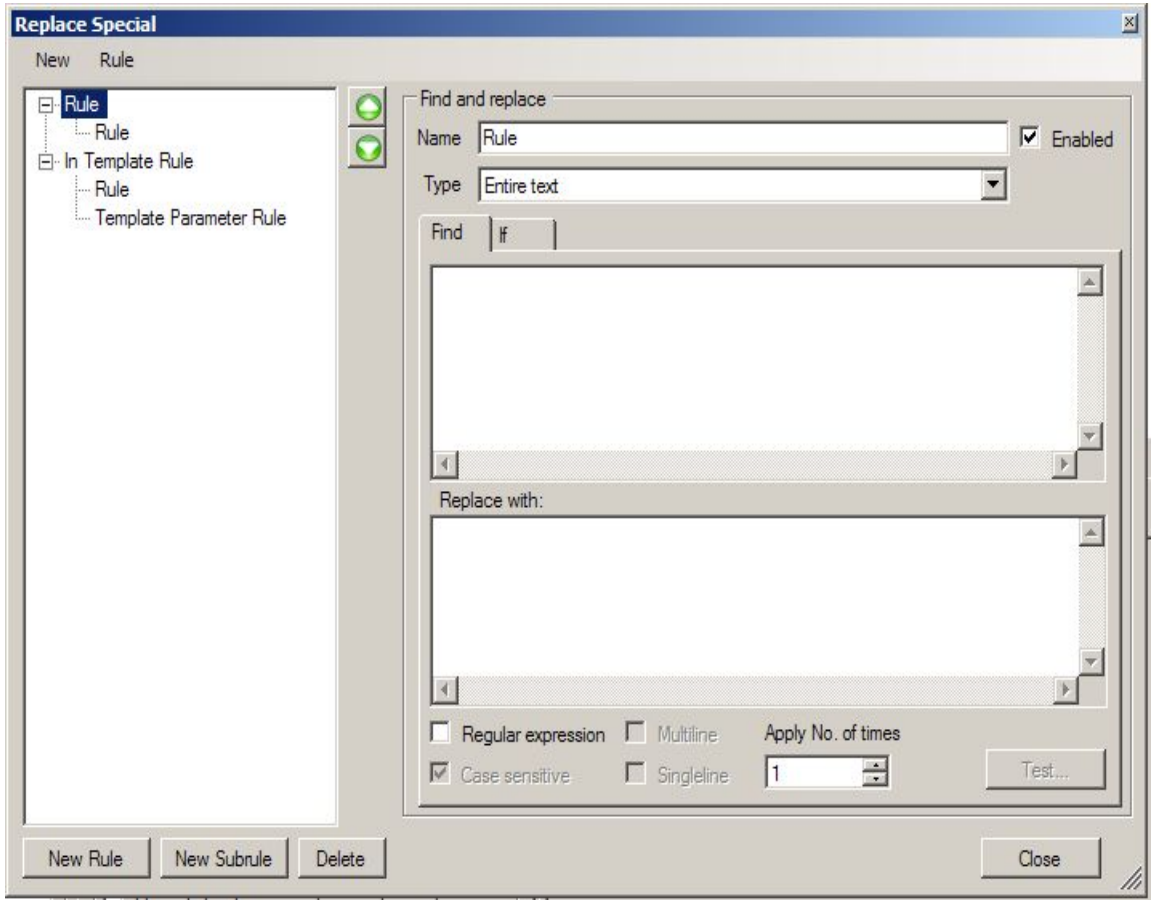
Advanced tools



Advanced tools



Advanced tools



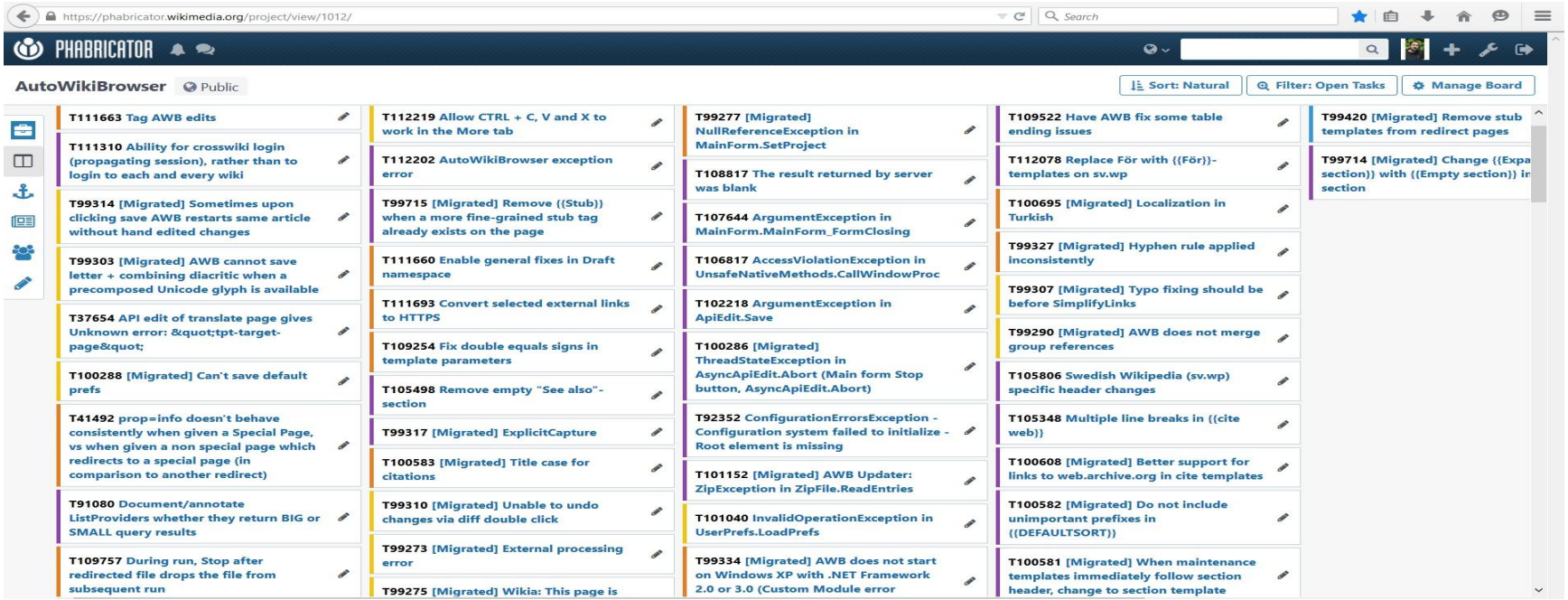
The team

https://commons.wikimedia.org/wiki/File:AWB team @ Wikimania 2013 5.JPG (Own work) [CC BY-SA 3.0 (http://creativecommons.org/licenses/by-sa/3.0)], via Wikimedia Commons

Phabricator

- It is a suite of web-based software development collaboration tools, including the Differential code review tool, the Diffusion repository browser, the Herald change monitoring tool, the Maniphest bug tracker and the Phriction wiki.
- Phabricator integrates with Git, Mercurial, and Subversion. It is available as free software under the Apache License, version 2.
- Phabricator was originally developed as an internal tool at Facebook. Phabricator's principal developer is Evan Priestley. Priestley left Facebook to continue Phabricator's development in a new company called Phacility.

Report bugs and ideas!



Acknowledgments

- The speaker received a WMF grant to travel and participate in this Conference
- Slides are available on commons