VisualEditor & Parsoid
Update, 2012-12-06
Our overall objective

- We are building a visual editor for MediaWiki (so it’s like a word processor)
- This will mean that editors don’t have to learn wikitext to edit articles
- Opening up editing to people who are dissuaded by the technical wiki code
- Freeing up editors to spend more time on *what* they write rather than *how*
What we expect to see

Users will (probably):

• make edits more quickly (time from clicking edit to saving will go down)
• bail on edits less (clicking edit, then browsing away will go down)
• make fewer edits that need to be reverted for breaking the page
• make fewer small follow-up edits (“fixes”) to the same page
How it works

Two major elements - the VisualEditor itself and Parsoid
Need to make sense of wikitext

''''The Stranger'''' is a [[novel]] by [[Albert Camus|Camus]].

Wikitext is great, but we really need structure!
Parsoid: wikitext ↔ HTML5

an HTML5-based content model and runtime for MediaWiki

```
'''''The Stranger''''' (''[[French language|French]]''): 'l'<nowiki />''Étranger'''') is a [[novel]] by [[Albert Camus|Camus]].
```

```
<b><i>The Stranger</i></b> (<i><a rel="mw:WikiLink" href="/French_language">French</a></i>): <<b><i>Étranger</i></b> (</i>) is a <a rel="mw:WikiLink" href="/Novel">novel</a> by <a rel="mw:WikiLink" href="/Albert Camus">Camus</a>.
```

```
The Stranger (<i>French: l'Étranger</i>) is a novel by Camus.
```

Browser  Visual Editor  Bots, caches etc
Challenges for Parsoid

Parsing the **bugs** features of wikitext

- overlaps: `<b>bold</b> <i>and italic</i>` only italic
- fragments: `{{table start}} {{table row}} {{table end}}`
- bug-for-bug compatibility ;-)

Round-tripping

- need to support wikitext storage and editing, **no dirty diffs**
- many ways to produce *the same* HTML: `[[Foo|bar]]`, `{echo` `[[Foo|bar]]}`, `[[{{echo|Foo}}|bar]]`, `[[...]]`, ...

Performance

- need to preserve and infer much more information
- PHP very slow for memory-heavy tasks, so used JavaScript (and will likely move to C++)
What we have learned so far

Any wikitext content can be represented as HTML + RDFa

- We enrich the HTML with semantic RDFa information
- Private (purely syntactic) data for round-trips kept in data-parsoid

A well-defined HTML5-based document model is very powerful

- Enables easier viewing/editing/processing using widespread tools
- Opens up new opportunities for optimizations like fragment caching

No hope without testing

- roughly doubled the number of parser test cases
- round-trip testing on 100,000 random pages from en:Wikipedia
We have run roundtrip-tests on **99782** articles, of which

- **99.99%** parsed without crashes
- **98.05%** round-tripped without semantic differences, and
- **79.94%** round-tripped with no character differences at all.

Latest revision:

- **Git SHA1**: 291a3cac60ff296b42fc1f388401ccd2001b883a
- **Test Results**: 11711
- **Regressions**: 145
- **Fixes**: 246

Averages (over the latest results):

- **Errors**: 0
- **Fails**: 0.04
- **Skips**: 2.26
- **Score**: 132.35
How it works

VisualEditor takes Parsoid’s HTML and lets users edit, then sends back
What are we doing now?

• On Tuesday 11 December we will launch VisualEditor on a few Wikipedias
• This will be an opt-in demo of the current version for feedback from users
• This is for editors to get familiar, highlight bugs, and help us prioritise
• It won’t replace the wikitext editor, but will let you edit & save articles
• This will get updated in our regular fortnightly cycle hereafter
Caveats

• Right now it is not ready for people to use as their regular editor
• Lots is not yet done - categories, langlinks, templates, tables, references, etc.
• Currently bits of pages render rather differently from read tab
• It will be much (much) slower to load editing pages than the final version
• For now, browser support sucks (Chrome and Firefox only); VE’s next focus
What it looks like

With the preference set on, you get an extra “VisualEditor” tab on articles.
What it looks like

With the preference set on, you get an extra “VisualEditor” tab on articles
What it looks like

Once in the editor you can change the text and save...
What it looks like

Some elements (images, templates, references) can't be edited yet.
What it looks like

“Native” browser editing means you get things like spell-checking and IME.
What it looks like

“Native” browser editing means you get things like spell-checking and IME.
What it looks like

Saving the page via Parsoid is easy ... and then you’re done
What it looks like

Saving the page via Parsoid is easy ... and then you’re done
What is to come in the future?

- Parsoid will be extended so that pages look the same as current read pages
- Parsoid will be ported to C++ so it’s faster, and can be the default parser
- VisualEditor will support editing of complex content
  
  Priorities: references, common templates, categories, images, table layout
- VisualEditor will work in more browsers (esp. Internet Explorer & mobile)
- We will expand deployment so we can get feedback from more editors
Questions?
Thank you for your time
Please feel free to ask us anything, any time