Let’s Completely Change How Templates Work!

C. Scott Ananian, Wikimedia Foundation
Wikimania Stockholm, 16 August 2019
Wikitext Templates are the best thing to happen to Wikipedia
“Templates, or custom messages, have grown from humble beginnings as an afterthought in a localisation feature. They are now used in almost 10% of pages in the English Wikipedia database. The 1.3 release brings several changes in recognition of this new role...”


2004-08-11
Wikitext Templates are the Worst!

So let’s change everything about them.
Some bad things.

- Surprising start-of-line rules
- Inconsistent quoting
- Unstructured inputs
- Unstructured output
- Mixing code/data/layout
- Problematic localization
- See “Templates are Dead!”, Wikimania 2015
What to do?

"Today's sunrise" by andreas_koenig is licensed under CC BY-NC-SA 2.0
1. Incremental fixes
2. Alternative mechanisms
3. Deeper changes
Incremental Fixes
Incremental 1: Balance

```html
<table>
<tr>
<td class="foo"> Hello </td>
<td> <b>world</b> </td>
</tr>
</table>
```
Incremental 1: Balance

```html
<table>
<tr>
<td class="foo"> Hello </td>
<td> <b>world</b> </td>
</tr>
</table>
```
Incremental 1: Balance

Opt-in with {{#balance}}
- {{#balance:block}}
- {{#balance:inline}}

Tidy to enforce balance
Prevent leaky templates
Increased parse/render speed

T114445
Incremental 2: Heredoc

{{tablestart|class="shiny"}}
| Hello || wiki = x
{{tableend}}

{{table|class="shiny"|
{{!}} Hello {{!}}{{!}} wiki &#61; x
}}
Incremental 2: Heredoc

{{tablestart|class="shiny"}}
| Hello || wiki = x
{{tableend}}

{{table|class="shiny"|}}
{{!}} Hello {{!}}{{!}} wiki &/#61; x
{{!}}

WIKIMEDIA FOUNDATION
Incremental 2: Heredoc

```
{{tablestart|class="shiny"}}
| Hello | wiki = x
{{tableend}}

{{table|class="shiny"|<<<
| Hello | wiki = x
| >>>}}
```
First person plural pronouns in Isthmus-Mecayapan Nahuat:

"nejamēn" ({{IPA|[nehameːn]}}) "We, but not you" (= me & them)
"tejamēn" ({{IPA|[tehameːn]}}) "We along with you" (= me & you & them)
First person plural pronouns in Isthmus-Mecayapan Nahuat:

: "nejamēn" {{IPA|[[nehameːn]]}} "We, but not you" (= me & them)
: "tejamēn" {{IPA|[[tehameːn]]}} "We along with you" (= me & you & them)
Incremental 2: Heredoc

More regular escape syntax
Makes it easier to write balanced templates
Perhaps can be further extended:

* a <<<
  multi
line
>>> list item

T114432
Incremental 3: JavaScript

Very briefly: add JavaScript as a Scribunto engine, alongside Lua

Pro: increase uniformity of codebase, leverage worldwide familiarity with JavaScript
Alternative mechanisms
Alternative 1: Glossaries

Word-based conversion dictionaries can get large and accumulate topic-specific entries.

Chinese wikipedia uses templates and a custom gadget to add new conversion rules.

Improve support by bringing this into core as article glossaries.

```
[[Glossary:Harry Potter]]

-{en-uk:Sorceror; en-us:Philosopher} -
-{en-uk:Minister for Magic; en-us:Minister of Magic} -
-{en-uk:sherbet lemon; en-us:lemon drop} -
```

T484
Alternative 2: Reduce!

Instead of using templates for everything, perhaps look for ways to remove templates?

- Replace Infobox/Navbox templates with
  - General header/footer mechanism?
  - Specialized dynamic builder extension?
- Instead of {{citation needed}}, {{subst:afd1}}
  - build specialized tools for certain tasks.
  - Community Tech team, original Flow idea
Alternative 3: Annotations

First person plural pronouns in Isthmus-Mecayapan Nahuat:

:"nejamēn" ({{{IPA|[nehameːn]}}}) "We, but not you" (= me & them)
:"tejamēn" ({{{IPA|[tehameːn]}}}) "We along with you" (= me & you & them)

Maybe this shouldn’t be crammed into a template at all?
Alternative 3: Annotations

Come to my talk
“A General Annotation Service”
Sunday 1230!

Also [[en:User:cscott/Ideas/Amazing Article Annotations]]
Deeper Changes
Deeper 1: Global templates

- Reduce duplication among wikis using a system similar to how “instant commons” works for media
- Implementation started with “Shadow namespaces” ([T91162](#))
- Ran into difficulties and is currently stalled (but there is a bot which copies templates between wikis)
- Main problem is....

[T121470](#)
Deeper 2: Template l10n

What needs to be translated?

● All parser functions, magic words, wikitext keywords and flags
● Name of template, names of template parameters, any template keywords or flags, TemplateInfo & documentation
● Scribunto modules, variable names, comments, documentation

First attempt: Yuri’s Multilingual Templates and Modules
Deeper 3: Visual Templates

I love you, but we need to break up.

Presentation != Code != Data

- Code should manipulate data or structured DOM (not wikitext strings)
- String data is automatically escaped
  - Your function can return “<a>” or “[[Foo]]” and it will appear literally
  - Return $(‘<a>’) or something similar if you want to emit a link
- Data is JSON or a wikidata query, not a wikitext string
- Presentation is editable with VisualEditor
  - Minimal markup, eg Spacebars variant of Handlebars
  - Structured semantics, not string concatenation

T114454
Data: Wikidata

**property names in the data context**

<table>
<thead>
<tr>
<th>event</th>
<th>gold</th>
<th>silver</th>
<th>bronze</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shotput</td>
<td>USA</td>
<td>GER</td>
<td>MEX</td>
</tr>
<tr>
<td>Discus</td>
<td>MEX</td>
<td>USA</td>
<td>GER</td>
</tr>
<tr>
<td>Javelin</td>
<td>GER</td>
<td>MEX</td>
<td>USA</td>
</tr>
</tbody>
</table>
Transformation of data context

Additional block helpers

Code: Scribunto/JS

```javascript
export default function(events) {
  // events is an array; each item has event, gold, silver, and bronze props
  var totals = new Map(), countries = new Set();
  function mget(m, key, defaultval) {
    if (m.has(key)) {
      return m.get(key);
    }
    return defaultval;
  }

  function inc(country, color) {
    var old = mget(totals, country, new Map(), color, 0);
    totals.get(country).set(color, old + 1);
    countries.add(country);
  }

  events.forEach((e) => {
    inc(e.gold, 'gold');
    inc(e.silver, 'silver');
    inc(e.bronce, 'bronce');
  });

  var rows = Array.from(countries).sort((a, b) => {
    var aa = totals.get(a), bb = totals.get(b);
    var c = aa.gold - bb.gold;
    if (c===0) { return c; }
    c = aa.silver - bb.silver;
    if (c===0) { return c; }
    c = aa.bronce - bb.bronce;
    return c;
  }).map((c) => totals.get(c));

  export function caption() { return document.createTextNode("caption"); }
```
### Handlebars-style placeholders

<table>
<thead>
<tr>
<th>Country</th>
<th>Gold</th>
<th>Silver</th>
<th>Bronze</th>
</tr>
</thead>
<tbody>
<tr>
<td>{#each row}</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>{country}</td>
<td>{gold}</td>
<td>{silver}</td>
<td>{bronze}</td>
</tr>
<tr>
<td>{#each}</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
An aside: Multilingual programming languages

“Coding Is for Everyone—as Long as You Speak English” (Wired) presents a very short list of multilingual programming languages:

- **Scratch** (I’d add **eToys** as well)
- **Blockly**
- **Excel formulas**
- **Wikitext**

Scoped variables and comments relatively easy to localize. OO APIs are harder; strong type system makes things easier. *(Perhaps Scribunto/TypeScript might be just enough type system to identify, and thus localize, invoked APIs, while still executing in a v8 runtime.)*
But wait there's more!
Three axes of change

How to invoke a template
- Here doc, Splat, etc

How to author a template
- Scribunto/JS
- Visual Templates
- Perhaps l10n and global templates too

How templates fit together
- Output data types
  - Wikitext string
  - (Balanced) DOM
  - Structured data
- Input data types
  - (Wikitext) string
  - Structured data
(T156876)
THANK YOU

cananian@wikimedia.org

[[User:cscott]]

WIKIMEDIA FOUNDATION