Wiki Migration

Dr. Vadim Zaytsev
(User:Spider, @grammarware)
© 2011
Outline

Wiki Migration

Motivation

7 slides

Community Migration

6 slides

Content Migration

Ontology Migration

9 slides

6 slides

http://www.rascal-ml.org/

http://grammarware.net/

http://www.rascal-ml.org/

Thank you!

Questions?
Motivation

7 slides
Wiki forking (community split)
Wiki forking (content split)
Wiki merging (true merger)
Wiki merging (acquisition)
Wiki moving (upgrade)
Wiki moving
Alternative wiki front-end
Community Migration

6 slides

*Persepolis* (2007), film by Marjane Satrapi and Vincent Paronnaud
Wiki community

✓ System operators & bureaucrats
✓ Technical administrators (and wikignomes)
✓ Active content generators (exopedians)
✓ Active content beautificators (wikifaeries & wikiimps)
✓ Discussion junkies (metapedians)
✓ Supplementary content creators
✓ Novice wikipedians (wikipuppies)
✓ Dormant participants (wikiangels & wikidragons)
✓ Wiki content readers (wikiprincesses)
Community roles

✓ Admin != Admin
✓ Citizen, Editor, Officer (Citizendium)
✓ SysOp, Bureaucrat, Check User, Editor (…)
✓ Warden, Expert, Master, Manager (WikiReality)
✓ Expert, Master, Warden (CycloWiki)
✓ SysOp, Bureaucrat, Check User, Editor, Reviewer, Moderator, Abuseditor (Lurkmore)
✓ …
Migrating community roles (1/2)

✓ Technical staff
✓ needs to be re-educated
✓ documentation & links
✓ Exopedians
✓ provide maximum information
✓ be as helpful as you can
✓ WikiFaeries and WikiGnomes
✓ give time to embrace new place & concepts
Migrating community roles (2/2)

- Metapedians
- value your time
- many issues to discuss
- Illustrators etc
- nice to have
- don’t be afraid to ask
- WikiPuppies
- make it easy
Community migration

✓ Re-registration?
✓ User information migration
✓ User statistics migration
✓ Passwords migration (imply trust in the admins)
✓ Name clashes
✓ Namespace clashes
Broader picture

✓ Search engine optimisation
✓ Informing the target audience
✓ Static links to your wiki
✓ Inside the wiki: portals, to-do lists, etc
✓ Integration
✓ Friends & sisters
Content Migration

9 slides

http://www.rascal-mpl.org/
Content migration

✓ Special:Export & Special:Import
✓ MediaWiki
✓ Wiki interchange format
✓ WS3, WIF, WAF, …
✓ Web scraping (HTML ⇒ Wiki)
✓ Wiki dialect mapping (Wiki ⇒ Wiki)
✓ Rendering and HTML importing
✓ TWiki, JSPWiki, …
Content migration methods

✓ Lexical approach
✓ one time effort
✓ very error prone
✓ Formal grammars
✓ good for programming languages
✓ Skeleton grammars
✓ more tolerant
✓ Semi-automated manipulation of structured content
## Wiki syntax: fonts/emphasis/…

<table>
<thead>
<tr>
<th></th>
<th>MediaWiki</th>
<th>TWiki</th>
<th>Wikidot</th>
<th>TiddlyWiki</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bold type</strong></td>
<td>&quot;&quot;text&quot;&quot;</td>
<td><em>text</em></td>
<td><strong>text</strong></td>
<td>&quot;text&quot;</td>
</tr>
<tr>
<td><strong>Italics</strong></td>
<td>&quot;text&quot;</td>
<td><em>text</em></td>
<td>//text//</td>
<td>//text//</td>
</tr>
<tr>
<td><strong>Bold italics</strong></td>
<td>&quot;&quot;&quot;&quot;text&quot;&quot;&quot;&quot;</td>
<td><strong>text</strong></td>
<td>///text///</td>
<td>///&quot;text&quot;///</td>
</tr>
<tr>
<td><strong>Monospace</strong></td>
<td>&lt;tt&gt;text&lt;/tt&gt;</td>
<td>=text=</td>
<td>{{text}}</td>
<td>{{text}}</td>
</tr>
<tr>
<td><strong>Monospace bold</strong></td>
<td>&lt;tt&gt;&quot;text&quot;&lt;/tt&gt;</td>
<td>==text==</td>
<td>{{<strong>text</strong>}}</td>
<td>{{{&quot;text&quot;}}}</td>
</tr>
<tr>
<td><strong>Underline</strong></td>
<td>&lt;u&gt;text&lt;/u&gt;</td>
<td>&lt;u&gt;text&lt;/u&gt;</td>
<td><strong>text</strong></td>
<td><strong>text</strong></td>
</tr>
<tr>
<td><strong>Strikeout</strong></td>
<td>&lt;s&gt;text&lt;/s&gt;</td>
<td>&lt;s&gt;text&lt;/s&gt;</td>
<td>--text--</td>
<td>--text--</td>
</tr>
</tbody>
</table>
## Wiki syntax: links/images/…

<table>
<thead>
<tr>
<th></th>
<th>MediaWiki</th>
<th>TWiki</th>
<th>Wikidot</th>
<th>TiddlyWiki</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wiki link</td>
<td>[[Target</td>
<td>text]]</td>
<td>WikiWord</td>
<td>[[[Target</td>
</tr>
<tr>
<td>External link</td>
<td>[URI text]</td>
<td>URI</td>
<td>[URI text]</td>
<td>[[text</td>
</tr>
<tr>
<td>Image</td>
<td>[[File:W.png</td>
<td>…]]</td>
<td>URI</td>
<td>[[image W.png]]</td>
</tr>
<tr>
<td>Table</td>
<td>{...}</td>
<td>{...}</td>
<td>{...}</td>
<td>{...}</td>
</tr>
</tbody>
</table>
Concrete wiki syntax migration

✓ EASY: Extract, Analyse, Synthesize.
✓ Extract = parse wiki pages
✓ write your own grammar
✓ use one of the existing grammars
Concrete wiki syntax example

module MediaWiki

syntax WikiText = WikiSymbol *;
syntax WikiSymbol = MWWord w | WikiLink wl;
syntax WikiLink
    = simple: "[[" PageName destination "]]
    | complex: "[[" PageName destination "]" | " MWWord+ description "]];

lexical MWWord = ![\t\n\r\[\]]+ !>> ![\t\n\r\[\]];
lexical PageName = ![#<>\[\]|{}\t\n\r]+ !>> ![#<>\[\]|{}\t\n\r];
layout MediaWiki = [\t\n\r]* !>> [\t\n\r];
Concrete wiki syntax migration

✓ EASY: Extract, Analyse, Synthesize.

✓ Analyse
  ✓ correctness validation
  ✓ data mining and inference
  ✓ dealing with ontologies

✓ Synthesize
  ✓ pretty-print wiki text
Concrete wiki syntax example

```
public str ppWT((WikiText) `<WikiSymbol * wss>`) {
return "<for(ws <- wss){><ppWS(ws)><}>";
}

public str ppWS((WikiSymbol) `<MWWord w>`) {
  return "<w>";
}

public str ppWS((WikiSymbol) `<WikiLink wl>`) {
  if ( wl.prod.def.name == "simple" )
    return "[[<wl.destination>]]";
  else
    return "[[<wl.description>|<wl.destination>]]";
}
```
Ontology Migration

6 slides
3.1 Wiki data model

What is in a wiki? Several definitions are possible, but we go for a definition from the end-user point of view. The end-user has two ways of interacting with wiki content: she can either browse the hypertext version or create and change content of pages using wiki syntax. It is the content written by users that has to be migrated to other wiki engines. Some important wiki features such as “backlinks” are inferable from the user-entered content. Arguably, usage data such as “10 most accessed pages” or the “recently changed pages” contribute considerably to the wiki usability. This data cannot be manipulated directly by users through the web interface. So we have to ask ourselves:

Wiki Page

User

Attachment

Section

List

Table

Paragraph

Link

Page Graph

Page Content

Table 1. A high-level view on the wiki data model
Wiki ontology: MediaWiki

What is essential?
Wiki ontology: TiddlyWiki
Ontology mismatches

✓ Article: current content or history of revisions?
✓ Media: attachments, files, articles?
✓ Authorship: implicit, collaborative, explicit, none?
✓ Tagging: recursive, folksonomous, predefined?
✓ Illustration: framed singles, galleries, inlines, none?
✓ Level of detail: many small articles, several lengthy?
✓ User information: how easy to get? how personal?
Stilistical mismatches

✓ Domain specificity
✓ “in the fictional universe X, …”
✓ Wikification
✓ aggressive or minimalistic
✓ POV
✓ NPOV or expert opinion
✓ Proofreading or a lack thereof
✓ Notability, OR, …
Conclusion

✓ Prepare well for migration
✓ Inform the community
✓ Automate content migration
✓ Resolve ontology mismatches
✓ Minimise problems for wikipedians
✓ Encapsulate new content
✓ Proceed with the new wiki!
http://grammarware.net/

http://www.rascal-mpl.org/

Questions?

Thank you!