Coding Tips for Wiki Sister Sites

Ward Cunningham
Cunningham & Cunningham, Inc.

The original WikiWikiWeb software has been cloned and modified many times over. In *The Wiki Way* Leuf and Cunningham describe how to modify their canonical implementation under the assumption that wiki site operators will want custom capabilities at their site. Each site then becomes a locus of custom that must be respected in any scheme to join sites into a larger information space.

Clones have always been encouraged. Clones may be cloned sites or cloned software or both. One idea that has carried through all clones is that of a distinguished link format for page names residing in the same database as opposed to those links resolved by the whole internet.

> WikiName -- resolved by wiki itself
> http://c2.com/index.html -- resolved by the internet

Several shorthand versions of the second format have been suggested as support for inter-wiki linking. We take the position that this does not lead to accidental linking which is the fortuitous connecting of ideas caused by intentional compression of the name space. We favor cross site linking based on the first form. We suggest establishment of Sister Sites, like Sister Cities, where continual effort is expended by site operators in order that accidental links can happen between their respective pages. A site becomes a sister site when the local site operator does three things:

> 1. Tracks the sister site's namespace, typically downloading
> a complete list of page names nightly.
>
> 2. Cites the sister site whenever a local page name duplicates
> one held by the sister site.
>
> 3. Renders page names as if they exist even when they
> only exist on a sister site.

This document describes each of these requirements in turn with examples drawn from our own experience with two versions of wiki.

1. Tracking Names

We now host all our wiki sites using cgi scripts that render pages stored as flat files named by their corresponding wiki name. We currently host a 20,000 page site on a small pc running Linux. We've found flat files to be portable, robust and fast. We attribute this to effective caching of directory pages within the operating system. We exploit this same mechanism for looking up page names from sister sites. We do so by creating zero length files named again by corresponding wiki names. Our typical wiki/4 file hierarchy looks like the following.

> `localsite` -- holds cgi scripts
>   `pages` -- holds flat files for each local page
>   `sisters` -- holds directories for each sister site
>   `sister1` -- holds metadata about sister site
>   `pages` -- holds zero length files for each sister page
>   `sister2` -- same structure as sister1

Each sister(i) directory contains these metafiles in addition to the pages subdirectory.

> `link` -- contains one line, a generic http link to the sister site
> `logo.gif` -- a graphic that will be used to cite pages on the sister site

When sister sites are served from the same server it is possible to track sister site page names by creating the sister(i) pages subdirectory as a symbolic link to the pages directory of the sister site itself.

When sister sites are served by remote servers it is necessary to connect to that server, retrieve a current list of page names, and then add or remove zero length files from the sister(i) pages subdirectory. The Portland Pattern Repository wiki offers a current list of page names at this url:

> `http://c2.com/cgi/wikiNames`

These requests are handled by this cgi script:

```
#!/bin/bash
echo "Content-type: text/plain"
echo
ls pages
```

The next page lists the program, load.pl, that updates wiki/1 sister page directories at c2.com, which are stored in a slightly flatter file hierarchy than that above.

```perl
#!/usr/bin/perl

# load.pl -- remote load sister site page names
# (c) 2001, Ward Cunningham

my $link = "[A-Z][a-z]+([A-Z][a-z]+)+";
&update ("wikibase", "http://c2.com/cgi/wikibaseNames");
&update ("why", "http://clublet.com/c/c/why?text=pages");
&update ("greencheese", "http://www.greencheese.org/AllPages");

sub update {
        my ($site, $url) = @_;
        my $dir = "$site.d";

        open (T, "TERM=vt100; lynx -source $url|") or die($!);
        my @new = grep (chomp, <T>);
        @new = grep (/$link/, @new);
        close (T);
        if (!@new) {
                warn("No data from $site at $url\n");
                return;
        }

        opendir(D, $dir);
        my @old = grep (/$link/, readdir(D));
        close (D);

        my (%new, %old);
        map($new{$_}++, @new);
        map($old{$_}++, @old);

        for (@new) {
                next if $old{$_};
                open (F, ">$dir/$_") or die ($!);
                close (F);
        }

        for (@old) {
                next if $new{$_};
                unlink ("$dir/$_") or die ($!);
        }
}
```

We run the load.pl script every night at 1:15 am. We use the Linux cron utility to launch the program based on the following crontab entry.

```
15 1 * * * (cd cgi/wiki.idb; ./load.pl)
```

The crontab format expects tab separated fields, the first five of which determine the periodicity by which cron will invoke the command specified in the sixth and final field. Linux users can edit their per user crontab with the command:

```
crontab -e
```

Cron collects output to standard error and, if any, mails it to the crontab owner. Load.pl is silent when things go well. It reports trouble should any site fail to return a non-empty page list. Occasional remote system outages create occasional email. Persistent trouble merits an inquiry directed to the sister site operator.

2. Citing Sister Pages

Wiki sites generally translate wiki pages to html on the fly with each request. As part of this translation the cash of sister site page names is examined to look for duplications of the the requested name. Here is the wiki/4 subroutine that performs this query.

```
sub Sisters {
    my ($title) = @_;
    my $result;
    for (<sisters/*/pages>) {
        if (-e "$_/$title") {
            s/pages$/link/;
            open (E, $_);
            my $link = <E>;
            close (E);
            $link =~ s/\$page/$title/;
            s/link$/logo.gif/;
            $result .= "<a href=$link><img src=$_></a>";
        }
    }
    return $result;
}
```

This subroutine takes a page name as argument ($title) and constructs a possibly empty string containing image icons hyperlinked to the corresponding pages on the

sister sites. The routine loops through all available sister site page directories identified by perl's unique file globing:

```
<sisters/*/pages>
```

This construct produces an array of file paths similar to the unix command:

```
echo sisters/*/pages
```

The results of the Sisters subroutine are included in the generated html output at some appropriate spot. Both wiki/1 and wiki/4 are happy to render non-existent page names and will cite extant sister pages for these too, if any.

3. Rendering Wiki Links

Wiki translators recognize wiki names in the page text and translate them to html hyperlinks called *anchors*. This is the subroutine wiki/4 uses to construct the anchor tag. It generates links of one form if the page exists, either locally or remotely, and another form otherwise. The second form is usually a question mark (?) linked directly to the edit page that would create the missing page.

```
sub AsAnchor {
    my($title) = pop(@_);
    my $isLocal = -e "pages/$title";
    # note -- need array context to exhaust the glob
    my @isRemote = <sisters/*/pages/$title>
                                  unless $isLocal;
    ($isLocal or @isRemote)
        ? "<a href=wiki.cgi?$title>$title<\/a>"
        : "<a href=edit.cgi?$title>?<\/a>$title";
}
```

Again we are using a file glob to query our database of page names:

```
<sisters/*/pages/$title>
```

This glob will be evaluated for every page name present in the text of the requested page that isn't already defined locally. We've tried to imagine schemes that would be more efficient than this but all are many times more complex and suffer from all kinds of startup or connection overhead.

The AsAnchor subroutine returns hyperlinks to the local wiki even when the pages are known to be at a remote sister site. An optimization here would be to generate a reference directly to the remote page so long as there is only one. We skipped this optimization because we wanted to give authors a chance to define the local page if that is their wish. Here is an example of wiki/4 rendering a non-existent page.

Notice that it offers an iconic link to the sister site as well as a button that offers to create a similarly named page locally. If the page were defined on more than one sister site then here is where the reader would choose which to visit first.

The original implementation of sister sites was content to display its usual default empty page message that went like this:

Describe ExtremeProgramming? here.

This seemed to be the prevailing instruction and distracted most readers from the subtle sister site citations.