Handling Relationships in a Wiki System

Advisor
Dr. Chris Pollett

Committee Members
Dr. Robert Chun
Mr. Auston Davis

By
Yashi Kamboj
Agenda

- Introduction
- Categorization in Wiki Systems
- Problem Statement
- Preliminary Work Summarized
- Relationship Handling System
- Experiments – Use Cases and Testing
- Areas of Improvement
- Conclusion
- Demo
About the Project

• This project adds a Relationship Handling system to Yioop, an open source content management system to enable relationship specification between the linked wiki pages.

• This allows binary representation of relationships, which does not exist in present day wiki systems.

• This system uses a novel approach by taking into account the linkages between wiki pages and facilitating search functionality on the basis of these links.
Introduction

• Wiki software enables users to manage content on the web, and create or edit web pages freely.

• Most wiki systems have a feature which allows pages to be grouped together as *categories*.

• This feature assists readers to navigate around related pages to fetch information corresponding to any subject area.

• Categorization is used in many of the enterprise wiki applications, such as Confluence, TWiki and SharePoint, as well as open source-applications like MediaWiki and XWiki.
Categorization in Wiki Systems

- Wiki systems support categorization of pages in a very traditional way by specifying whether a wiki page belongs to a category or not.

- The central goal of the category system is to provide navigational links to all Wikipedia pages in a hierarchy of categories which readers, knowing essential—defining—characteristics of a topic, can browse and quickly find sets of pages on topics that are defined by those characteristics.

- Categorization represents unary relationship and is not sufficient to represent n-ary relationships, those involving links between multiple wiki pages.

- To create a category, add [[Category:Parent category name]] to MediaWiki of the wiki page.
Categories are organized as overlapping "trees", formed by creating links between inter-related categories. Partial view of Wikipedia’s category system is shown above.
As we have just seen, Wiki systems have a convoluted structure, which is difficult to manage and retrieve useful information.

Currently, Yioop specifies linkages between wiki pages but the type of relationship type that binds any two pages is missing.

Relationship Handling system has been added to Yioop to allow Wiki structure to be explored efficiently, made easily searchable, and navigable on the basis of relationship type between the Wiki pages.

Use cases for this feature include genealogy, ontology, and dependency identification.
When categorization was initially added to wiki systems in 2004, there was no mechanism to limit the search results for a particular category. Very large categories caused performance issues.

Later changes were made to limit the search results to 200 entries per page. Again, for large categories, users had to navigate through multiple pages in order to see all the results. Thus, page by page mechanism also proved impractical.

In mid-2005, the category table of contents was introduced. With the table of contents, it became possible to navigate through large categories with a few clicks.
• The categorization of Wiki systems has both been analyzed and visualized.

• Holloway et al. [1] compared the top categories and classification structure of Wikipedia 2005 to widely used encyclopedias.

• A more recent study by Kittur et al. [2] analyzed the growth of categories and developed an algorithm to map wiki articles to the top 11 chosen categories.

• Although we see that significant research has been done in the field of wiki categorization structure, the exploration of n-ary relationships is a topic that did not attract much attention among the various studies done on the structure of wiki systems.
Multiple approaches have been proposed to implement categorization in Wiki systems. These are as follows:

- Intersection
- Manual addition
- Pseudo namespace
- Field-value pairs
- IEG proposal using Metrics
The problems associated with mentioned approaches are as follows:

- Impractical
- Prone to Errors
- Making changes to MediaWiki without including relationship type
- Problem with pages belonging to very large categories
- Inconsistent results, dependent on metrics
This system works by fetching the type of relationships that link any two wiki pages.

It stores all relationship types in the database and facilitates transitive relationship exploration and search functionality on the basis of relationship type.

This feature implements binary relationship between wiki pages, which is sufficient to represent n-ary relationships.

We combined our approach of adding relationship types linking wiki pages to the approach discussed in the previous slide (Field-value pair), thus, making changes to the wiki markup language syntax.
• In Wiki markup language, free links can be created by putting double square brackets around text designating the title of the page that needs to be linked to.

• Thus, [[California]] will be rendered as California. Optionally, a vertical bar (|) can be used to customize the link title.

• For example, typing [[California|The Golden State]] in the wiki markup will produce a link that is displayed as The Golden State but, in fact, links to California.

• Working on similar approach, relationship type can also form part of media wiki syntax when specifying links between wiki pages as follows: [[capital|Sacramento|Click to go to capital city]] will be rendered as Click to go to capital city and will take the user to Sacramento’s wiki page, saving relationship type as capital between California and Sacramento.
“What Links Here” feature

• Wiki Parser is responsible for parsing MediaWiki documents, both within Yioop, and when Yioop indexes MediaWiki dumps as from Wikipedia to an HTML equivalent.

• The wiki parser of Yioop fetches each individual MediaWiki internal links present in MediaWiki documents to be able to discover connections between different pages.

• This is done using a regular expression. For example, if page A is related to page B with a link that is displayed as Page B Link, then MediaWiki of document A would have [[PageB|Page B Link]].

• Thus, the presence of these types of internal links can be extracted using a regular expression. Once we have all the links between the pages, it can be saved to the back end, and later used for information retrieval of linkages between different pages.
“What Links Here” feature

- Consider a test group, named ‘NewTest’ with three Wiki pages: Main, test1new, and test2new. The links between them are summarized as below:

<table>
<thead>
<tr>
<th>Relationship Type</th>
<th>From Page</th>
<th>To Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>parent</td>
<td>Main</td>
<td>test1new</td>
</tr>
<tr>
<td>parent</td>
<td>Main</td>
<td>test2new</td>
</tr>
<tr>
<td>parent</td>
<td>test1new</td>
<td>test2new</td>
</tr>
<tr>
<td>child</td>
<td>test1new</td>
<td>Main</td>
</tr>
<tr>
<td>child</td>
<td>test2new</td>
<td>test1new</td>
</tr>
</tbody>
</table>
“What Links Here” feature (Contd.)
“What Links Here” feature (Contd.)

![Image of Yoop! admin interface with links to test1new Page, test2new, and Main]
“What Relates Here” feature

- To delve further into how wiki pages link with each other, i.e., information about their relationship types, “What Relates To” feature is implemented.

- It works in conjunction with the wiki parser to extract the relationship type by creating a link extractor regular expression which takes into consideration the relationship type.

- For example, if page A is related to page B with a link that is displayed as “Page B Link,” and a relationship “Parent,” then MediaWiki of document A would have `[[Parent|PageB|Page B Link]]`.

- The fetched values, i.e., relationship type and the linked page are then stored in the database and can be displayed to check what type of relationships link these page to other pages.
“What Relates Here” feature (Contd.)

- Admin [Feeds and Wikis]

Account Access
- Manage Account
- Manage Users
- Manage Roles

Crawls

What relates to test1new Page

child
parent
“What Relates Here” feature (Contd.)

- Admin [Feeds and Wikis]

Account Access
- Manage Account
- Manage Users
- Manage Roles

Crawls

- Admin [Feeds and Wikis]

Account Access
- Manage Account
- Manage Users
- Manage Roles

Crawls

Main Page at NewTest Group
- Main
  - test1new
  - test2new
  - test1new Page at NewTest Group
“Advanced Search” feature

• The Group Page list in Yioop displays all the wiki pages present in a particular group. The advanced search feature has been added to it by displaying a search text box, which enables search on the basis of relationship type chosen and any particular page.

• The user can choose multiple relationship types and the results would display all wiki pages linked to that particular page with the chosen relationship type.

• These relationships can be further explored to check which pages link “from” and which pages link “to” the wiki page selected by the user.
“Advanced Search” feature (Contd.)

- Admin [Feeds and Wikis]

Account Access
- Manage Account
- Manage Users
- Manage Roles

Crawls
- Manage Crawl
- Manage Classifiers
- Page Options
- Results Editor
- Search Sources
- Web Scrapers

Social
- Manage Groups
- Feeds and Wikis
- Mix Crawls

NewTest Group Page List

Search group page titles [Go] [Advanced]

NewTest Group Page List

- Main
  Test new Main page!

- test1new
  This is test1 new page!

- test2new
  This is test2 new page!
“Advanced Search” feature (Contd.)
“Advanced Search” feature (Contd.)
Basic flow of processes in RH system
The experiments conducted on the Relationship Handling (RH) system give an idea about the working of the system under ideal conditions. Here, an assumption is made that the content added to wiki pages adheres to MediaWiki syntax.

These experiments are divided into four parts:

- Experiments conducted to test the working of all new features added to RH system – Part of Demo
- A/B Testing
- Performance Testing
- Experiments conducted on use-case scenarios of the RH system
Experiments – A/B Testing

- This testing, also known as split testing, involves comparing two different versions of a web page to see which one performs better.

- The two variants are shown to similar visitors and their responses are recorded. The one that provides a better conversion rate, wins.
The results of A/B testing are as follows:

A/B Testing

- A: 4.0
- B: 2.0
- None: 1.0
Experiments – Performance Testing

• This testing was done by adding multiple relationship links to a wiki page and then creating that page. The time of creation was noted for a wiki page by varying the number of links.

![Performance Testing Graph](image)
The developed system has been found really useful in the field of genealogy to organize the family tree structure better and understand it efficiently.

This system has been tested on Dr. Pollett’s family tree containing around 300 members.

Screenshots depicting the usefulness of this system to manage large family trees easily are as follows:
The Group Page List under Genealogy group displays all members of Dr. Pollett’s family tree.
To test Dr. Pollett’s relationship to other members of the family, search for “Christopher Pollett” in the search box.
Dr. Pollett’s Wiki page is as follows:

**Christopher John Pollett**

**Birth:** 1970-08-19

**Relationships**
- Son: Harry Fraser Lancaster Pollett
- Son: Judith Anne Whittaker Pollett
- Husband: Divorced Ronghui Xu
- Husband: Mary Ninh Pollett
- Brother: Allan Fraser Pollett
- Brother: Heather Anne Pollett Wesson
- Father: James Xu Pollett
- Friend: Ralph Doncaster

**Attachments**

**Details**
- 1991 Full Professor SJSU
- Got tenure 2001. Spring 2008 took leave working at Mozilla. Fall 2008 back to SJSU.
- 2001-2002 Asst. Prof., Computer Science, San Jose State University
- 2002-2006 Asst. Prof., Computer Science, San Jose State University
- 1999-2001 Adjunct Asst. Prof., Program in Computing, UC Los Angeles (Adjunct Asst. Prof)
- 1987-1998 Lecturer, Computer Science, Boston University.
- PhD in Math 1997 UC San Diego
- B.S. in Math 1990 UC San Diego
- B.S. in Math 1990 College
- Born Halifax, Nova Scotia.
Dr. Pollett is related to Mr. Harry Fraser Lancaster Pollett with the relationship son. His father’s wiki page can be explored further by clicking on the link displaying his name.
To check all people with whom Dr. Pollett is linked, choose “What links to Christopher John Pollett Page” from the drop down.
To check all relationship types that link Dr. Pollett are as follows.

- brother
- father
- friend
- generic_links
- husband
- mother
- sister
- son
- wife
Any relationship can be further explored to check “from” and “to” relations.
• The developed system can also be used to identify entities that have no dependencies or those entities that are not dependent on any other entities.

• This can be done using a crawler to figure out pages with no links “to” and “from” other pages.

• For example, a project with multiple components where execution of one component is dependent on another. We can find relationships between various components and figure out which needs to be executed first.
Areas of Improvement

• The areas of improvement includes scenarios where two-way relationship exists between two entities. For example, if A is father to B, then B is son/daughter to A.

• In this system, both relationships have to be specified separately. Improvements can be made where if one side of such relationships is specified, other sides get identified automatically.

• In addition to this, future scope includes creating category and sub category pages using the syntax: [[category name| Text to be displayed]] and creating multiple relationships to a wiki page simultaneously by using check boxes.
• In this project, a new module called Relationship Handling system is developed for Yioop.

• This module is responsible for storing the information in a manner such that it can be later used for understanding the links between wiki pages and searching based on relationship types.

• This system employs an approach of storing the relationship types and links between wiki pages in the database to be used for exploring relationships and facilitating search features.

• It is very useful to understand the convoluted structure of wiki systems and letting the user explore content easily and effectively.
References


