A COLLABORATIVE KNOWLEDGE BASE WITH RICHNESS, IMMEDIACY, AND RESPONSIVENESS

Russ Abbott
Department of Computer Science
California State University, Los Angeles
Los Angeles, California
USA
Russ.Abbott@Gmail.com

ABSTRACT
We describe a web-based collaborative knowledge base that offers immediate access to one's data (very few keystrokes), automatic and rich cross linking, wiki-like user generation and editing of data, an integrated database, an integrated attachments pool, and integrated discussions and journals.

KEY WORDS
Responsiveness, immediacy, automatic linking, pseudo links, integration.

1. Introduction

The Collaborative Knowledge Base (CKB) is a web-based system for working with a body of knowledge. It is intended for use by either an individual or a team. Like a well-designed website (it is web-based), it offers users immediate access to data and requires very little navigation.

Although CKB has a fairly large number of features, the features are all so well integrated and natural that it provides a uniform and easy-to-understand interface. CKB's primary inspirations are Wikipedia [1], Moodle [2], and the increasingly large number of powerful web content management systems (see, for example, [3]).

2. CKB Overview

A CKB consists of a collection of web pages divided into the following categories:

- **Topic** pages are pages, each of which is about some topic of relevance to the site. When used by an individual who is working on a book, for example, one would have Topic pages about the various topics in the book. When used by a team working on a project, one would have Topic pages for the various project components and sub-components. One Topic page is designated to be the site's **Home** page.
- **News** pages are Topic pages that contain items of immediate but possibly transient interest. Although not all CKB sites will have News pages, it is likely that most will. Summaries of the most recent News pages are displayed in the navigation panel.
- **Tables** are database tables. Since CKB is built on top of a database, it is possible integrate a basic database functionality into it with little extra effort.
- **Views** are queries into the CKB Tables.
- **Calculators** are pages in which predefined spreadsheet-based calculations are defined. This feature is Microsoft-specific and works only on Internet Explorer. Any Excel spreadsheet may be saved in a form that allows it to be displayed interactively on a web page. Changes to data elements may be made, and revised calculations are performed, but the changes cannot be stored. Hence the term *calculator* rather than the more general *spreadsheet*.
- **Journals** are pages in which users may keep personal observations. They function very similarly to blogs.
- **Calendar** pages are events in a calendar.
- **Help** pages are Topic pages that contain the system's Help information.
- **Additional categories** Additional categories of pages may be defined by users as needed. If, for example, there is a need for a collection of Management pages, such a category may be created by users—without having to ask a webmaster to reorganize the site.

In addition, for each page there is an associated **Discussion** page on which users may comment about the subject page.

Any page may also have **attachments**, files that have been uploaded. An attachment may be attached to one, many, or no pages. Since it is possible to request a list of all attachments, attach-

ments that are not attached to any pages are easily accessible.

Each attachment may have a user-written comment associated with it. When attachments are listed, the comment associated with the attachment is displayed. This tends to be much more informative than simply listing the name of an uploaded file.

3. Rich Linking

CKB includes a number of automatic linking capabilities that give the overall knowledge base a rich interlinked structure.

3.1 Auto-links

Each page may have a list of aliases associated with it. Whenever the title of a page or an alias of a page appears on another page, a cross-link is automatically generated from the referring page to the subject page. The summary of the subject page appears as a *tool-tip* when the mouse hovers over the link.

To avoid irrelevant links, it is possible to block any individual appearance of a term from acting as a link.

3.2 Pseudo-links

When the user drags the mouse over a word or phrase, the selected word or phrase is sent as a query for a Google search. When the user double-clicks on a word, the word is sent as a query to OneLook [4] (www.onelook.com), a website with links to a large range of dictionaries. The effect is as if each word is linked to its dictionary definition, and each word or phrase is linked to its Google search page.

Like much of the CKB functionality, the user could accomplish these lookups themselves with more work on his or her. The effect of the pseudo-link feature, though, is to further the sense of close integration among information resources.

3.3 New Windows and Same Windows

When clicked, internal links, i.e., those generated by automatic cross-linking, open in the same window. Links to pages outside the CKB (including the pseudo-links) open in a new window. Attachments also open in a new window. This helps the user develop a sense of the CKB as a context with an inside and an outside.

4. Tables, Views, and User Controlled Structures

4.1 See-through Hierarchies

If a CKB is being used for subject matter that naturally divides into a few major sub-areas, category hierarchies may be defined over the set of pages. Category hierarchies are called *see-through* or *transparent hierarchies*. At each level of the hierarchy, all the elements below that level are visible. Thus at the top of the hierarchy, all pages are visible. As one descends in the hierarchy, fewer pages nodes are visible. This allows one to see everything in the hierarchy from the top level (in case one forgot where a particular element was placed), but it also allows categorization and structuring of pages in the system.

4.2 Database Tables and Views

Since, as indicated, CKB is built on top of a relational database system, it made sense to offer users a fairly complete, if basic, database capability. Users may create tables, populate, and modify tables (both the structure and the content) through a simple, intuitive GUI.

Users may also query their tables using standard SQL queries. Queries may be saved as views. Since defining a View requires a knowledge of SQL, it is the most technically demanding feature of a CKB's user interface. To make defining Views as easy as possible, a form is provided in which the user enters the individual SQL clauses separately.

4.3 Using tables to control the hierarchy

CKB exploits its user-oriented database capability by storing the user-level page hierarchy in a user-accessible table. Thus users may modify the structure and organization of their pages simply by editing a relational database table..

5. User Generated Content

The left panel of every page contains a list of the page categories, a search field (which is similar to the Google search in that it allows *all terms, some terms,* and *excluded terms*), and the headlines and summaries of the most recent News pages.

When the user clicks on a category name (or the word *Attachments*), the summaries for all the existing pages in that category (or of the Attachments) are displayed along with a links to those pages (or Attachments). The user may also click an entry at the top of the list that allows him or her to create a new page in that category (or upload a new Attachment).

User input of this sort appears *in-place* on the page. It is as if the pages "opens up" and a form in which the user enters the name and summary of the new page appears. (Technically, this is done by modifying the value of the display attribute from none to inline.)

The effect is quite pleasing; one avoids the sense that it is necessary to go to a different page to enter information about the new page. One simply and naturally types information about the new page directly into a page that is already listing information about all the other pages in that category.

After entering information about the new page, a button sends the information to the server, which generates the new page and returns it to the user for more content.

All elements of a page are user-editable. One can change the title, the summary, the main content, or the attachments directly. (The comment associated with an attachment may be changed wherever the attachment appears.) New attachments may also be added to pages directly. As when entering information about a new page, user editing of a page's content also occurs *in-place*: the user clicks an edit icon and again the page "opens up" and allows the user to edit the text.

CKB uses an open-source WYSIWYG web editor [5], which both enhances the sense of immediate access to the content and helps the user maintain a sense of working in context: he or she has the sense of simply editing the page as one would when working on a document using a word processor.

All pages are archived so that if one later decides to erase changes, or if a change had been done maliciously, previous versions of a page my be restored—again by the user, who can examine the previous versions and simply pick one.

6. Subscribing to Pages

Every page has a *subscribe-to-this-page* button—which becomes an *unsubscribe-to-this-page* button when the user is already subscribed to the page.

This features lets CKB provide in an integrated manner the sort of functionality that normally requires multiple separate systems.

- If one subscribes to a Topic, Table, or View page, one is informed about any changes to that page. This allows one to remain informed about changes that others make to the page.
- If one subscribes to an index page (index pages are the pages generated when the user asks for all pages in a category or all attachments), one is informed whenever a new page in that category is generated, when a page in that category is deleted, or when a new attachment has been uploaded.
- If one subscribes to a Discussion page, one is subscribing to what amounts to a forum or bulletin-board discussion about the subject pages associated with the Discussion page. In addition subscribers to Discussion pages are sent the new entries as they are added. Although not yet implemented, it will eventually be possible to add entries to a Discussion page by email.
- If one subscribes to a Journal page, one is informed whenever a new entry is added to the Journal. As with Discussion pages, the new entry is sent to all subscribers. It should eventually be possible to update Journal pages by email. Journal pages may thus be used as distribution lists.

  The primary difference between a Discussion page and a Journal page is that a Discussion page is intended to be about another page on the system. A Journal page, like a blog, is intended for entries by an individual user or group of users.

6.1 Page Permissions

Only certain users may be authorized to make changes.

7. Interface Uniformity

CKB presents a consistent interface to users even in situations in which the operations it performs are quite different. Topic pages, Table pages, Discussion pages, Journal pages, and View pages are implemented though very different mechanisms. But those differences are hidden from the user, who is presented with a uniform interface.

The same is true when subscribing to pages. It takes very different mechanisms to allow a user to subscribe to a Topic page, for example, than to subscribe to an index page, i.e., a page that lists, for example, all Topic pages. Yet as far as the user is concerned, the process (clicking the *subscribe-to-this-page* button) and more important the meaning of subscribing is identical in both situations—notifications of changes are sent to the user. A great deal of work has been done to provide a simple consistency for services that are similar but significantly different.

7.1 Plug-and-Play

CKB's interface uniformity is built on an internal structure that supports the addition of new features

in a plug-and-play manner. Pages, Tables, Views, Calendar Events, and other kinds of pages are treated as page types, each of which has its own implementation of the standard CKB user-level operations. For example, displaying a topic page requires different operations from displaying a database Table. From the user's standpoint, however, the two are accessed (and edited) in a similar manner. New types of pages may be integrated into the system simply be implementing the appropriate API operations.

7.2 Design Goals: The Three I's

CKB's primary design goals are Integration, Intuitive Design, and Immediacy.

Integration. Most if not all of the features offered by CKB are available in other systems. CKB makes these features available to users in an integrated framework in which the features appear just when you want them. The page subscription features is a good example. It provides, a foruming capability without having to go to a special forum or bulletin board system.

Intuitive Interface. Users find that CKB offers a natural interface that does what they expect. There is a very shallow learning curve. Most of what users want is immediately visible in the way that users expect it to be. Even features that one normally doesn't think about are done intuitively. An example is the fact that links between pages within a CKB open in the same window, and links to outside pages open in a new window.

Immediacy. Information is presented to the user with minimal navigation. This distinguishes CKB from document storage systems in which most of the important information is not visible until one navigates to and then opens a document. Like most web sites, most information is immediately presented to the user on the page at which he is looking.

Besides the direct web-oriented presentation of information, two other examples of this sort of immediacy are (a) the fact that uploaded documents (attachments) may have brief descriptions associated with them (so that one need not open a document to see what it contains) and (b) CKB's transparent hierarchies and search capabilities so that one rarely finds oneself frustrated thinking that some information was stored somewhere but one can't remember where.

8. Conclusions

CKB integrates into one system a number of user interface features that are found on a diverse set of systems. In this context, the term *user interface* refers to all interfaces to a system, including email notifications, and not just to interfaces that one uses when in direct contact with a system.

CKB's primary operational goals are to provide data immediacy and intuitive access to information in a framework that both (a) facilitates the creation of richly interconnected information components—i.e., links that make a collection of information modules into a knowledge base—and (b) makes it easy for multiple users to collaborate on creating, editing, and extending such a knowledge base.

9. Acknowledgements

Thanks to Yong Jin who made the WYSIWYG editor and web calendar work and provided much other support.

REFERENCES

[1] Wikipedia., http://www.wikipedia.org.

[2] Dougiamis, M., Moodle: a software package for producing internet-based *social constructionist* courses and web sites, http://moodle.org/doc/.

[3] ContentManagementMatrix. http://www.cmsmatrix.org/

[4] OneLook, http://www.onelook.com/.

[5] InteractiveTools, HtmlArea, http://www.interactivetools.com/products/htmlarea.