Next-Generation Wikis: What Users Expect; How RDF Helps

Axel Rauschmayer

Institut für Informatik, LMU München,
Oettingenstr. 67, 80538 München, Germany
`http://hypergraphs.de/`

Abstract. Even though wikis helped start the web 2.0 phenomenon, they currently run the risk of becoming outdated. In order to find out what aspects of wikis will survive and how wikis might need to evolve, the author held a survey among wiki users. This paper argues that adding RDF integration to wikis helps meet the requirements implicitly contained in the answers of that survey. Technical details are given by looking at the semantic wiki Hyena.

Key words: Next-generation wikis, Semantic Wiki, RDF, Semantic Web

1 Introduction

Wikis have been a popular web application for some time now. But the recent rise of Ajax [1] has changed the perception of web applications: Many wiki-like web sites have appeared, often specialized to do a single task well (where wikis are more universal). Examples include Google Calendar and Flickr.

This begs the question: What aspects of wikis will survive in the Web 2.0 age? What aspects are worth preserving? To answer this question, one has to find out what people use wikis for, what features they like and what features they are missing. A small survey on this topic helped the author do that. This paper interprets the survey results as requirements for the next generation of wikis and then argues that wikis that mix wiki pages and RDF data are perfectly equipped to meet those requirements. *How* they can meet those requirements is illustrated by taking a closer look at the semantic wiki Hyena [2].

2 The survey

The survey has intentionally been relatively simple. For example, it was probably not representative for all potential wiki users, because the participants (a total of 23) were chosen in an ad-hoc fashion by announcing the survey to the author's colleagues[1] and to a mailing list about semantic wikis. But the results are still interesting and point out several possible trends for wikis.

[1] None of them are experts in the semantic web or wikis.

The survey participants answered in percentages indicating how important a given fact was to them or how often they performed a given activity. The reported percentages are averages of these answers. Note that the answers do not necessarily apply to a single wiki; many participants use several wikis. The following sections present groups of questions and observations that the author derived from the answers.

2.1 What is a wiki used for? What is its content?

Collecting data or knowledge	91%
Coordination, planning, project management	56.75%
Web site, light-weight content-managment system	54.5%
Document creation (and later publishing)	48.75%
Discussions, forum	42%
Brainstorming, (possibly shared) whiteboard	39.75%
Weblog, relatively small journal-style entries	16%

“What is the purpose of your wiki? What do you use it for?”

The content of traditional wikis is just text. But what users care about is the data and knowledge contained in the text—as expressed by the 91% ranking of “collecting data or knowledge”.

Text	68.25%
Data	55.75%
Knowledge	55.75%

“What makes up the content of the wiki?”

In the survey, “text” was explained as feeling more like a word document, possibly being a collection of notes. “Data” are lists, tables, forms, etc.—things one might keep in a spreadsheet or a database. “Knowledge” is similar to data, but the focus is on collecting facts (true statements) and on specifying these facts as precisely as possible. Thus, the numbers above confirm what all the tables and lists in traditional wikis already suggested: In addition to text (semi-structured data, if you will), structured data and knowledge play an important role when it comes to wiki content. Naturally, structured information could be more flexibly processed if it were explicitly stored and had dedicated editors. For example, spreadsheets handle tabular data well, so it would be nice if one could embed little spreadsheets inside a wiki page. Note that the survey results do not indicate that wikis should become pure databases.Rather, being able to mix text and data is what seems to make wikis attractive.

2.2 Who uses the wiki?

Several collaborators, all reading and writing	60.25%
Personal use, a single person	50%
Few editors, many readers	46.5%

At heart, wikis can be considered groupware. Still, having the wiki information available anywhere and the flexibility in structuring information, makes wikis good personal information managers: Survey participants attributed an average importance of 50% to this task.

2.3 Current and future wiki features

Information roaming: the wiki information is available online.	78.5%
Collaboration: share and jointly edit information.	77.25%
Linking: relate and collate pieces of information.	72.75%
Publishing: disseminate information.	67%

"What core aspects of (traditional) wikis are you interested in?"

Version control (editing history, who edited what, unlimited undo, etc.)	78.5%
File upload and management	69.25%
Wiki page meta-data (annotations and labels describing the content of the page)	58%
WYSIWYG text editor	56.75%
Generate a PDF file from a wiki page	52.25%
Diagrams (UML, mind maps, organizational charts, etc.)	48.75%
Finer-grained wiki pages	47.75%
Outliners (edit indented lists such as tables of contents)	46.5%
Live collaborative editing (all editors work on the same copy of the document, changes show up immediately)	46.5%
Spreadsheets (with calculation)	46.5%
Discussions (forums)	41%
Offline editing, synchronization	37.5%
Calendars	37.5%
Form-based data entry (similar to MS Access)	37.5%
Blogs	25%

"What (actual or hypothetical) features are important to you?"

The author thinks that while an offline mode has been ranked relatively low, it is still essential for next-generation wikis. Otherwise, information will not be truly available everywhere (1st table); especially for personal information management (Sect. 2.2), one will need to access it without online connectivity.

2.4 Wiki alternatives

The following is a list of web applications that the survey participants use as alternatives to wikis for some tasks: (1) BackPack, (2) Blogger, (3) del.icio.us, (4) Facebook, (5) Flickr, (6) Google Calendar, (7) Google Docs, (8) iusethis, (9) Online Contacts, (10) Trac, (11) Wordpress, (12) WikipediaReview.com.

Interestingly, the majority (all except 1, 4, 11) of these web applications is very task-specific. Accordingly, Sect. 2.3 indicates that users would like to see more task-specific editing support (including WYSIWYG text editors) in wikis. The difficulty is to do so without significantly raising the learning curve.

3 Hyena

Hyena is an RDF publishing and editing system that comes in two components: A desktop application (Eclipse plugin, Fig. 1) and a web application (Java Servlet, Fig. 2) for online editing.

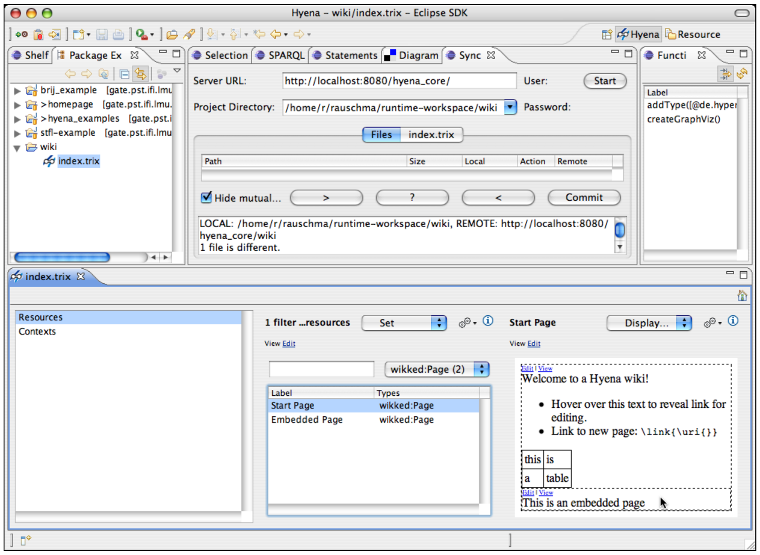
1. Storing and editing data: RDF is used as universal data storage. Hyena supports a variety of data encoded in RDF resources and has specialized graphical editors for them. Working with RDF generates *presentation data*: Lists of resources returned by a query, bookmarked locations, etc. This data can be edited in a similar fashion to RDF resources and saved (*manifested*) as RDF data.
2. Integrating pages and data: Wiki pages are also stored as RDF resources. They an link to external data or embed it. Similar to data-specific editors, embedding is supported by data-specific *embedders* (translators from the data to the abstract wiki syntax). All of a page's references (links, embeddings, etc.) to RDF resources are made explicit in RDF. This prevents stale links and allows one to track referers.
3. Meta-data for pages: Every page being an RDF resource, it can be annotated with RDF. This meta-data can be referenced in a query whose results can be manifested and embedded (as a table, as a sequence of embeddings, via templates). Thus, pages and data can be collated and presented in many ways.
4. Online and offline availability, collaborative editing: Both the desktop application and the web application manage web sites as *projects*, directory trees with files. This includes images, shared files and RDF data. One can synchronize projects between the desktop and the web. For RDF data, synchronization granularity is resources, otherwise it is files. Thus data can be published to a web server, but also edited offline. Furthermore, projects are easy to back up (which was one of the explicitly mentioned wishes in the survey).

The requirement of integrating text and data (Sect. 2.1) is fulfilled by items (1) and (2). The desired features "page meta-data" and "offline editing" (Sect. 2.3) are provided by items (3) and (4). Task specific editing (Sect. 2.4) is explained in item (1). For more detailed information on Hyena, consult [2]

Acknowledgments Thanks to Malte Kiesel, Andreas Schroeder, Philip Mayer, and Hubert Baumeister for their feedback on the survey questions.

References

1. J. J. Garrett. Ajax: A new approach to web applications. `http://www.adaptivepath.com/publications/essays/archives/000385.php`, 2005.
2. A. Rauschmayer. Wikifying an RDF editor. 2007. Submitted for publication.



Server URL: http://localhost:8080/hyena_core/ User:

Project Directory: /home/r/rauschma/runtime-workspace/wiki Password:

Path	Size	Local	Action	Remote

☑ Hide mutual...

LOCAL: /home/r/rauschma/runtime-workspace/wiki, REMOTE: http://localhost:8080/
hyena_core/wiki
1 file is different.

Label	Types
Start Page	wikked:Page
Embedded Page	wikked:Page

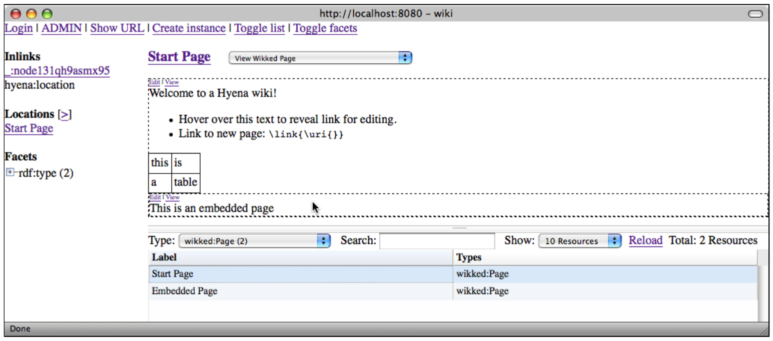
Welcome to a Hyena wiki!

- Hover over this text to reveal link for editing.
- Link to new page: `\link{\uri{}}`

this	is
a	table

This is an embedded page

Fig. 1. The Eclipse frontend for Hyena. The bottom half shows three editing panes that have been populated from left to right and thus make the history of editing visible.The top left is one of the standard Eclipse file explorers. One can see that the file `index.trix` that is being edited below belongs to project `wiki`. The top center hosts a variety of views that supplement the editor at the bottom. Currently visible is a view for synchronizing Eclipse projects with a web server. The top right contains a view that shows what commands can be used in the current context.



Welcome to a Hyena wiki!

- Hover over this text to reveal link for editing.
- Link to new page: `\link{\uri{}}`

this	is
a	table

This is an embedded page

Type: wikked:Page (2) Search: Show: 10 Resources Reload Total: 2 Resources

Label	Types
Start Page	wikked:Page
Embedded Page	wikked:Page

Fig. 2. The same project `wiki` that has been edited with Eclipse in Fig. 1 is displayed here as a web application in Firefox. The bottom right half shows the *resource list*, a subset of all available resources that is determined by filter criteria (such as "show only wiki pages", which is currently active). Resource `Start Page` has been selected and is visible in the top right. To the left, there is a toolbar that shows inlinks (resources that point to the currently selected resource); a list of special locations (the first of which is shown by default when starting the web application); and *facets*, a subset of all property key-value pairs among the resources in the resource list. Facets are grouped by key. One can hide the facets and the resource list so that Hyena/Web feels more like a web site.