Semantic authoring and retrieval within a Wiki

David Aumueller

Department of Computer Science, University of Leipzig
Augustusplatz 10/11, Leipzig 04103, Germany

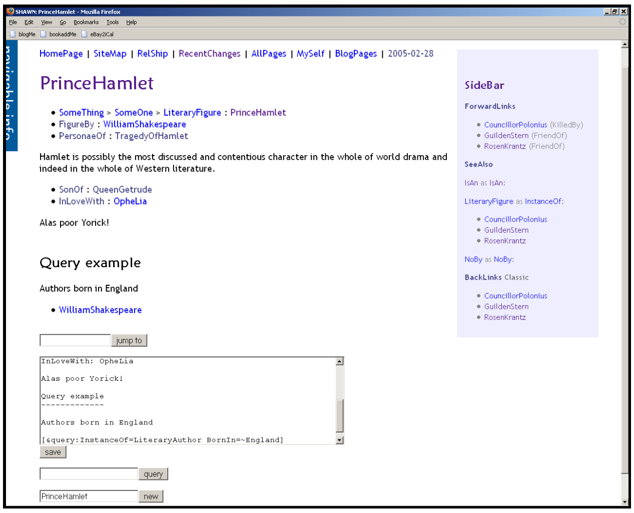
david@informatik.uni-leipzig.de

ABSTRACT

Structuring information on the Web can be a tedious task but is needed for effective retrieval of specific content. A prototype web application – the SHAWN Wiki – helps the user in this task by facilitating semantically structured content creation in an effortless and usable manner. Semantic authoring merely is a matter of entering field value pairs among prose text within a single input field. Annotating the content that way the user instantaneously gets gratified by context aware means of navigation and semantic query possibilities on an emergent ontology. Queries can be persistently embedded on Wiki pages to maintain constantly up-to-date lists of results. Thus, the information authored on the Wiki remains highly accessible.

1. INTRODUCTION

Authoring content should be possible in a straight-forward manner to not distract the writer from her stream of thoughts. The ease of publishing documents on the World Wide Web culminates in Wiki environments [1]. On Wiki driven sites users may author content on the web site itself without even knowing HTML. The downside of this quick approach is that specific information is hard to find and the user is quickly lost in an abundance of Wiki pages. Usually, all pages reside within one conceptual level, woven together by mere hyperlinks that result from specially formatted words, so called CamelCase or WikiWords, i.e. names made up by concatenating capitalized words. Following such a Wiki link leads to a Wiki page with that name.


SHAWN: PrinceHamlet - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
blogMe bookaddMe eBay2iCal
navigable info
HomePage | SiteMap | RelShip | RecentChanges | AllPages | MySelf | BlogPages | 2005-02-28
jump to
save
query
new
PrinceHamlet

PrinceHamlet

- SomeThing > SomeOne > LiteraryFigure : PrinceHamlet
- FigureBy : WilliamShakespeare
- PersonaeOf : TragedyOfHamlet

Hamlet is possibly the most discussed and contentious character in the whole of world drama and indeed in the whole of Western literature.

- SonOf : QueenGetrude
- InLoveWith : OpheLia

Alas poor Yorick!

Query example

Authors born in England

- WilliamShakespeare

```
InLoveWith: OpheLia

Alas poor Yorick!

Query example
-------------

Authors born in England

[&query:InstanceOf=LiteraryAuthor BornIn=~England]
```

SideBar

ForwardLinks

- CouncillorPolonius (KilledBy)
- GuildenStern (FriendOf)
- RosenKrantz (FriendOf)

SeeAlso

IsAn as IsAn:

LiteraryFigure as InstanceOf:

- CouncillorPolonius
- GuildenStern
- RosenKrantz

NoBy as NoBy:

BackLinks Classic

- CouncillorPolonius
- GuildenStern
- RosenKrantz

Figure 1. Interface of the SHAWN Wiki prototype.

This paper presents an effortless approach to author semantically structured content on a Wiki. Assuming each Wiki page resembles a (real world) concept, arbitrary relationships between concepts of any kind can be modelled. This relationship data is entered on Wiki pages in a straightforward and usable manner. The resulting metadata instantaneously gets used to benefit the user. Firstly, the user gets gratified by context dependent means of navigation on the Wiki. Secondly, using simple semantic queries the user can find specific content or collect up-to-date content to her personal likings by embedding queries on Wiki pages.

The next section explains how to enter metadata on the Wiki, how the user benefits from this in terms of navigation, how the Wiki can be semantically queried, and how external applications may work in conjunction with the Wiki. Section 3 outlines the current implementation of the prototype and its shortcomings.

2. FEATURES OF SHAWN

A web page displayed by the presented Wiki prototype named SHAWN (Structure Helps A Wiki Navigate) is divided into three parts, as illustrated in Figure 1: The text input/edit box, the rendered Wiki text above, and the sidebar on the right hand side containing context-dependent links, constructed using information from other pages.

2.1 Effortless authoring and semantic editing

The SHAWN prototype succeeds with only a single text input field to enter both prose and metadata at the same time. Common approaches to authorizing semantically structured content force the user to enter data specifically into multiple input fields, each field designated for a specific purpose, e.g. [2]. In SHAWN entered field value pairs – simply formatted with a colon in between – get interpreted as data triples. In terms of RDF (Resource Description Framework) where information is held as triples containing subject, predicate, and object/literal, the page name of a Wiki page denotes the subject of a triple; the field name being a WikiWord denotes the predicate; the value denotes the object or literal, depending on whether the value resembles a Wiki page itself or just a literal value. Thus, on a page named "PrinceHamlet" the line "FigureBy: WilliamShakespeare" gets interpreted as the triple *<PrinceHamlet> <FigureBy> <WilliamShakespeare>*. Using the URI of the Wiki as base URI these triples can be exported to fully valid RDF statements, the building blocks of the Semantic Web.

2.2 Context dependent means of navigation

The entered metadata immediately gets used to display relevant context dependent navigational elements. These are twofold. Firstly, breadcrumbs inform the user about her position in the Wiki showing the path back to the root of the site or concept.

Demos and Posters of the *2nd European Semantic Web Conference (ESWC 2005)*, Heraklion, Greece, 29. May – 1. June 2005

Secondly, links to pages that link to the current page using special relationship types get displayed in the sidebar. In Wikis generally, pages that link to the current page are referred to via *backlinks*. Since in SHAWN pages denoting e.g. a more specialized concept of another one contain an “is-a” declaration to the super concept, the backlink to the more specialized page can be seen as *forwardlink* leading deeper into the site or taxonomy. The breadcrumbs on the other hand show the way back, up the hierarchy. Thus, two important questions regarding usability of a web site get answered: “Where am I?” and “Where can I go?” [3].

Usability of the Wiki web site is further enhanced by using the metadata to automatically create a sitemap or any hierarchy of relationship types in use, a feature common unstructured Wiki environments lack. Surplus, the graph spawned by all connected pages could be interactively visualized using a Java applet, highlighting or filtering nodes by relationship type.

2.3 Semantic retrieval and persistent queries

Since the entered metadata are available as triples as outlined above, the Wiki site can be queried semantically, using a triple store that supports Semantic Web query languages [4]. Currently, queries in SHAWN are formulated as in “predicate=object” or “predicate=~literal” to find all concepts that satisfy the query, i.e. pages that have the questioned properties set to the questioned object or match the regular expression on the literal. Specifying the property can be omitted from the query to retrieve all triples that have any property set to the object or literal in question. Combinations concatenated by spaces to apply ‘and’-semantics are possible, among others. A complete syntax for the query interface on the Wiki got not yet devised, though.

The challenge of an effective semantic query interface is to offer the user high flexibility and usability at the same time, so that both novice and advanced users find what they are looking for. The first, often entering less specific queries might need to browse through a longer list of results, though. For example, entering only one construct of a triple (subject, predicate, or object/literal) will result in more hits and need to be viewed in context, i.e. the other two parts of the resulting triples should be returned; querying on two parts of a triple, only the remaining part needs to be returned.

SHAWN accepts a variety of commands to be embedded on Wiki pages to generate content gathered from all available data. E.g. the links in the sidebar get created by putting special operators in a template Wiki page. Likewise, queries in SHAWN need not produce only search results available temporarily to the current user but can be embedded persistently within Wiki pages via simple operators. Whenever a page containing a query operator gets loaded the specified query is executed and the result gets embedded seamlessly within the rendered text – a special *view* on the Wiki information space. That way it is e.g. possible to maintain pages containing always up-to-date query results. Newly entered content anywhere on the Wiki will instantly affect the result without having to reformulate the query. Making single Wiki pages available as RSS (Rich Site Summary/Really Simple Syndication) would allow the user to know immediately any changes to her interest without even visiting the Wiki.

2.4 Integration with external applications

Often, Wikis are used for personal information management (PIM), as suggested by [1]. Conceivably in this context are collections of Wiki pages that denote people one knows, containing contact data such as addresses and dates of birth or future appointments. These dates can be extracted from SHAWN and made available e.g. to the user’s calendar application where she gets automatically reminded of important events. The prototype extracts dates as iCalendar format (e.g. Mozilla). Users often maintain lists of favourite web resources, i.e. bookmarks and quotes of pages encountered while reading. These may be of interest to one person only or a whole community. SHAWN offers to collect bookmarks and passages from web pages to store for one’s own use in the PIM context or to make them available to a community. Using a Wiki to maintain bookmarks and web quotes makes them accessible not only from within the one browser application they got collected but from any web browser on any machine. Additionally, they can be annotated further.

3. CURRENT IMPLEMENTATION

The current prototype is implemented in Perl with each Wiki page stored as plain text file on the web server. These get transformed to XHTML and CamelCase terms get hyperlinked. The prototype relies on Unix ‘grep’ to extract the triples. As query engine RAP (RDF API for PHP) supporting RDQL queries is used at present, but the need for recursion to determine the transitive closure of specific relationship types (e.g. “is-a”) will eventually lead to using more complex implementations. Apart from using triple stores incorporating query engines supporting RDFS/OWL semantics, inference processors such as CWM could be plugged in to devise the missing triples before querying the model [4].

4. DEMONSTRATION

To demonstrate the SHAWN prototype users will be guided in authoring, navigating, and semantically querying the information space (e.g. beers/brewers), as well as embedding queries on Wiki pages. It will be shown how entered dates can be accessed via the Mozilla calendar application, and how arbitrary Web pages can be bookmarked and passages therein extracted into the Wiki.

5. CONCLUSION

The presented Wiki prototype SHAWN makes it easy to enjoy the benefits of structured data, a challenge identified by [5]. Within SHAWN it is straightforward to author both text and metadata. At the same time it is uncomplicated to query the collection of documents by keyword as well as semantically, using a simple query interface. Persistently embedding queries into Wiki pages allows monitoring very specific changes of content collected on the site. With SHAWN it is exceptionally easy to create triples for the Semantic Web. OWL vocabulary will be supported in the future to broaden the semantics available to queries.

6. REFERENCES

[1] Leuf, B., Cunningham, W.: The Wiki way: quick collaboration on the web. Addison-Wesley (2001)

[2] Tazzoli, R., *et al.*: Towards a Semantic Wiki Wiki Web. Proc. of International Semantic Web Conference (2004)

[3] Nielsen, J.: Designing web usability: the practice of simplicity. New Riders, Indianapolis, Ind. (2000)

[4] Furche, T., *et al.*: Survey over Existing Query and Transformation Languages. http://rewerse.net (2004)

[5] Halevy, A. Y., *et al.*: Crossing the Structure Chasm. Conf. on Innovative Database Research (2003)