Strata: Typed Semi-Structured Data in DokuWiki

Brend Wanders
University of Twente – Databases group –
Enschede, The Netherlands
b.wanders@utwente.nl

Steven te Brinke
University of Twente – Formal Methods and
Tools group – Enschede, The Netherlands
s.tebrinke@utwente.nl

ABSTRACT

A semantic wiki is a wiki that has a model of the knowledge contained in its pages. Currently, semantic wikis are not adopted by a large user base, because most implementations are research prototypes that implement their own wiki engine. To increase familiarity with semantic wikis and quick adoption of semantic technologies we present Strata, a plugin for the well known wiki DokuWiki. Strata allows the use of semi-structured data in any DokuWiki installation, normalizes values based on their types, and allows extensive data modeling and querying on complex data structures.

Keywords

DokuWiki, semantic wiki, SPARQL, structured data

1. INTRODUCTION

A wiki is an application that allows people to collaboratively add, modify, or delete content. Wikis have little implicit structure, allowing structure to emerge according to the needs of users. In general, wikis are centered around pages: content resides on a specific page. Due to this nature, it is hard to share content across pages. When multiple views on the same content are desired, commonly, this content is added multiple times to the wiki. Repeated content leads to duplication of work and easily introduces inconsistencies.

To tackle this problem, several wiki extensions provide so-called structured data, allowing (untyped) data to be used across pages. Whereas this avoids most inconsistencies, it still remains challenging to enter data consistently. When data entry is split across multiple pages, not all values might be entered in the same way. For example, if one event is listed on *27-08-2014* and another one on *27 August 2014*, both events have a different date when the values are compared literally. Therefore, it is beneficial to use data types, in order to see that both values represent the same date.

DokuWiki [5] is an open source wiki system. It is actively developed, with ease of installation and low system requirements in mind, and aims at being a good documentation and note-taking environment for small teams. All content in DokuWiki is strictly based on pages: all data is stored in text files. In our view, such a simple data model is good, because it allows users to easily edit content, since such content resides on a specific page.

In this paper, we present Strata: a structured-data extension for DokuWiki that normalizes values based on their types. Strata extends DokuWiki by allowing users to enter and query semi-structured data. Its uses range from automatically generating indices to extensive data modeling and querying on complex data structures.

2. WHY STRATA?

A semantic wiki is a wiki that has a model of the knowledge contained in its pages. Several implementations of semantic wikis exist: Semantic MediaWiki [6], IkeWiki [7], SweetWiki [1] among numerous others. Of these, only Semantic MediaWiki sees significant use on publicly accessible sites. Most of these implementations are research prototypes that implement their own wiki engine. This approach works well for the purpose of researching new methods and facilities. It works less well for the adoption of semantic wikis by the larger user base. By building on top of the well known DokuWiki, we hope to increase familiarity with semantic wikis and quick adoption of semantic technologies. The Structured Data plugin [4] is a previous effort that builds on top of DokuWiki, but it provides a much simpler data model than Strata, which lacks extensive query capabilities.

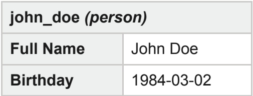
Strata augments the DokuWiki content model with a database that provides an index to create various views on the same data. The database contains the normalized forms of data derived from the wiki pages, so removing the database or our plugin does not remove any data from the wiki, it only removes some views on the data.

Data typing in Strata follows the same principle as structure in wikis: types are not strict and can be added to the needs of users. The idea is that when a data type is specified, the data is normalized to this type whenever possible and left unchanged otherwise.

For users of the wiki, Strata offers data entry with normalization based on types. This helps users by processing consistent and comparable normalized formats allowing the use of common notations with both data entry and querying. Queries can be formulated that combine data from several separate data entries, allowing the user to get a view of the combined data. Query results can be displayed as either table or list and can be sorted and filtered on the client. Sorting can be configured per case to work from left-to-right (the

OpenSym '14, Aug 27-29 2014, Berlin, Germany
ACM 978-1-4503-3016-9/14/08.
http://dx.doi.org/10.1145/2641580.2641636

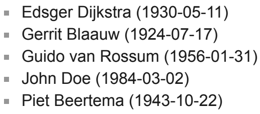
```
<data person>
Full Name: John Doe
Birthday [date]: 1984-03-02
</data>
```



john_doe *(person)*	
Full Name	John Doe
Birthday	1984-03-02

Figure 1: Data entry with ‘date’ type hint.

```
<list ?name ?birthday>
?p is a: person
?p Full Name: ?name
?p Birthday [date]: ?birthday
?birthday <= 1990-01-01
</list>
```



- Edsger Dijkstra (1930-05-11)
- Gerrit Blaauw (1924-07-17)
- Guido van Rossum (1956-01-31)
- John Doe (1984-03-02)
- Piet Beertema (1943-10-22)

Figure 2: Query to show a list of all people born before the 1st of January 1990.

default) or right-to-left in case this fits the data, for example lists of addresses such as *5 David Street* which should be sorted on street name before number.

For researchers and developers, the Strata plugin offers a good base for further experimentation. The plugin is designed with extension or modification in mind, allowing quick introduction of new types and query result views. Next to the default backend for SQLite, both MySQL and PostgreSQL are supported, and there is an interface for the implementation of support for other backends. The source code is well-documented and the plugin has been in use for over a year on a small but active wiki.

3. DATA ENTRY & QUERYING

Data is entered with dedicated wiki syntax. Type hinting is used to determine normalization and display format of entered data as seen in Figure 1. Entered data will be coupled to the page it is entered on. Fragment identifiers can be used to subdivide the entered data into smaller parts relating to different subjects. Most users intuitively see entered data as data about the page’s subject. The system attaches no special significance to the page on which data is entered.

After entering data, the user can query the data, an example of which can be seen in Figure 2. Queries are written in a query language based on SPARQL [3]. The query language is designed to match the syntax for data entry and to allow expressing simple queries in an intuitive way. The query language is converted to SQL for the underlying RDBMS as described by Chebotko et al. [2]. Type hints can be used in the query language to associate types with variables and literals. Literals within the query, such as a date, are normalized according to their associated types. Types are propagated through the variables. In the example in Figure 2 the literal in the comparison is automatically normalized as a ‘date’ due to the hinted type associated with ‘?birthday’.

It is our experience that most users formulate their queries in an ‘is this true?’ fashion. Query answering matches this expectation by collapsing multiple identical answers into a single answer. In effect, query answering is done using set semantics instead of SPARQL’s bag semantics.

The results of a query are displayed to the user in table or list format. Both formats support client-side filtering and sorting. Type hints from the query are used to determine appropriate display forms for the resulting values. Aggregates can be used on resulting values, for example to display a count or sum of values.

4. FREE-FORM CONSTRAINTS

Because of free-form nature of a wiki, the users should not feel constrained on what kind of data they want to enter. However, the data is more easily accessible and usable if it maintains at least some form of homogeneity. With Strata we wanted to support the user in maintaining this homogeneity, but we did not want to enforce a certain constraint format or representation on the user.

Users can draw up constraints in an advisory capacity: the violations will be reported, but users can still enter data that violates the constraints. In our vision, this increases the flexibility of the system, as users can handle the violations as they see fit. For example, they can revise the data later on or address the problem by updating the constraint itself.

By expressing constraints as queries, the user can easily create a view of all data not conforming to the constraints. Furthermore, expressing all or part of the data constraints by entering them as data on the wiki itself, the constraints queries can be generalized.

5. CONCLUSION

The Strata plugin presented in this paper allows the use of semi-structured data in any DokuWiki installation and as a basis for further work and experimentation.

During the demonstration at OpenSym 2014, we will show the flexibility of simple data entry and querying, as well as more complex examples of using data constraints. We will also demonstrate the client-side sorting and filtering interaction. We hope to gather feedback on the topics of user interaction and free-form constraint handling.

A. RELEASE

The full plugin has been released under a GPL license, and is available online on GitHub. More information is available on the Strata plugin page on the DokuWiki site.

Strata release on GitHub
`https://github.com/bwanders/dokuwiki-strata`

Strata plugin page
`https://www.dokuwiki.org/plugin:strata`

B. REFERENCES

[1] M. Buffa, F. Gandon, G. Ereteo, P. Sander, and C. Faron. SweetWiki: a semantic wiki. *Web Semantics: Sci., Services and Agents*, 6(1):84–97, 2008.

[2] A. Chebotko, S. Lu, and F. Fotouhi. Semantics preserving SPARQL-to-SQL translation. *Data & Knowledge Engineering*, 68(10):973–1000, 2009.

[3] S. H. Garlik, A. Seaborne, and E. Prud’hommeaux. SPARQL 1.1 query language. `http://www.w3.org/TR/sparql11-query/`.

[4] A. Gohr. Structured Data Plugin. `https://www.dokuwiki.org/plugin:data`, 2007–2014.

[5] A. Gohr and the DokuWiki Community. DokuWiki. `https://www.dokuwiki.org`, 2004–2014.

[6] M. Krötzsch, D. Vrandečić, and M. Völkel. Semantic MediaWiki. In *The Semantic Web-ISWC 2006*, pages 935–942. Springer, 2006.

[7] S. Schaffert. IkeWiki: a semantic wiki for collaborative knowledge management. In *15th IEEE Int. Workshop on Enabling Technol.: Infrastructure for Collaborative Enterprises*, WETICE’06, pages 388–396, June 2006.