# SHAWN: Structure Helps a Wiki Navigate

David Aumueller

Department of Computer Science
University of Leipzig
Augustusplatz 10/11
04103 Leipzig
Germany
david@informatik.uni-leipzig.de

**Abstract:** Common wiki applications lack possibilities to structure the relationships between wiki pages. This paper presents a semantic wiki prototype named SHAWN that allows modelling concepts and their relationships within a wiki environment. One goal of this prototype is to keep concept creation very simple. Yet, entering relationship data instantaneously gratifies the user with enhanced navigational means on the wiki. The engine supports simple semantic queries upon the emergent model. A challenge is to accommodate a self-explaining query interface for these ontologies.

## 1 Introduction

### 1.1 The Wiki Way

With the success story of Wikipedia, the online encyclopedia created by its users, underlying wiki engines become a hot topic. The content of wikis is edited online, on the wiki page itself. Wikis use simple markup rules to denote headlines, lists, emphasis, or image inclusion. Their main strength though lies in creating hyperlinks to other pages within the wiki. In the original wiki, the Portland Pattern Repository by Ward Cunningham, any so-called *CamelCase* words are interpreted as links to pages having the words as title. Following such a link opens the corresponding wiki page or just an edit box if the page does not yet exist. The user can go ahead and put in some content. Thus, lots of pages can be created quickly, but overview is lost even more quickly. **Wikis lack structure.** Usually, all the pages exist on merely one conceptual level, hyperlinks do not carry any semantics. The here presented prototype SHAWN introduces typed links to wiki paradigm. Thereon, technologies from the Semantic Web come into play.

## 1.2 The Semantic Way

The Semantic Web, a vision of the future Web, should give information on the current Web “well-defined meaning [to better enable] computers and people to work in cooperation” [1]. The common approach to creating semantically rich data, i.e. information annotated with metadata, is to export to Resource Description Framework (RDF) from existing well-structured data. RDF consists of subject-predicate-object triples that state specific facts about resources or concepts, e.g. “[Shakespeare] <isAuthorOf> [Hamlet]”, whereby subject, predicate, and object (if not a literal) are identified via Uniform Resource Identifiers (URI). Using RDF standardized relationship types and classes are created as so called vocabularies. The most prominent vocabulary is the Dublin Core Element Set [2] which offers a core of relationship types e.g. to identify typical metadata of publications, such as author and date.

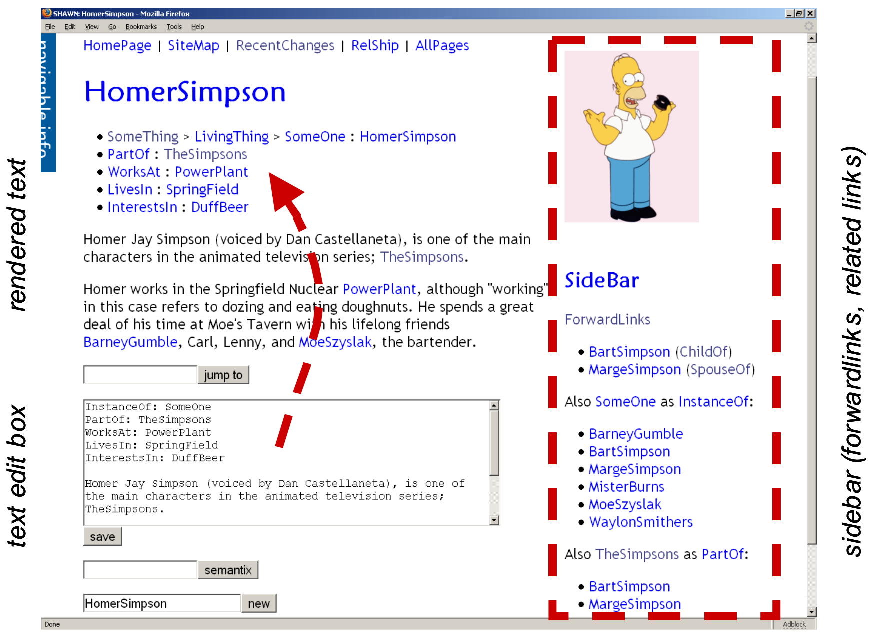
The top-down approach of creating semantic markup by exporting well structured databases prevails due to the fact that the opposite approach of creating semantically rich data bottom-up is tedious. With not even syntactically valid web sites one cannot expect their authors to write semantically marked up pages. It is the ease of publishing which made the Web the most successful type of media in the last decade. Its popularity emerged due to the relaxed interpretation of incomplete and invalid markup by web browser applications. Authors were instantaneously gratified by seeing their content published online.

## 1.3 The Wicked Way – SHAWN’s Graze

The wiki concept is accepted by one of the largest communities on the Web – Wikipedia.org would not be that successful and could not have published almost half a million articles in the English edition within the four years of its existence. The wiki way [3] is an appropriate means to entice ordinary people onto the Semantic Web via instant gratification [4].

The here presented prototype of the semantic wiki SHAWN wants to pursue those steps by offering the following key features within a wiki environment:

- **Effortless** editing of metadata to create semantic structure between wiki pages.
- **Instant** gratification by exploiting this structure for navigational aids.
- **Semantic** search and information retrieval.
- **Inferring** new concepts via simple reasoning.


rendered text
text edit box
sidebar (forwardlinks, related links)
SHAWN: HomerSimpson - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
navigable info
jump to
save
semantix
new
Done
Adblock

HomePage | SiteMap | RecentChanges | RelShip | AllPages

HomerSimpson

- SomeThing > LivingThing > SomeOne : HomerSimpson
- PartOf : TheSimpsons
- WorksAt : PowerPlant
- LivesIn : SpringField
- InterestsIn : DuffBeer

Homer Jay Simpson (voiced by Dan Castellaneta), is one of the main characters in the animated television series; TheSimpsons.

Homer works in the Springfield Nuclear PowerPlant, although "working" in this case refers to dozing and eating doughnuts. He spends a great deal of his time at Moe's Tavern with his lifelong friends BarneyGumble, Carl, Lenny, and MoeSzyslak, the bartender.

```
InstanceOf: SomeOne
PartOf: TheSimpsons
WorksAt: PowerPlant
LivesIn: SpringField
InterestsIn: DuffBeer

Homer Jay Simpson (voiced by Dan Castellaneta), is one of
the main characters in the animated television series;
TheSimpsons.
```

HomerSimpson

SideBar

ForwardLinks

- BartSimpson (ChildOf)
- MargeSimpson (SpouseOf)

Also SomeOne as InstanceOf:

- BarneyGumble
- BartSimpson
- MargeSimpson
- MisterBurns
- MoeSzyslak
- WaylonSmithers

Also TheSimpsons as PartOf:

- BartSimpson
- MargeSimpson

Figure 1: Screenshots of a wiki page within the SHAWN prototype

A page in SHAWN as in Figure 1 mainly consists of three parts: the text edit box, its resulting rendered text, and the sidebar containing links resulting from other pages. The text in the edit box is the only data that exists for the concept of *HomerSimpson*. Extracted relationship types get rendered with bullet points to confirm that the SHAWN engine recognized those triples. The property value pair `InstanceOf: SomeOne` in Figure 1 within the edit box (typeface Courier) gets transformed to the breadcrumbs "SomeThing > LivingThing > SomeOne : HomerSimpson", by merely having specified the immediate parent concept via an is-a property. The information in the sidebar stems from recognized triples within other wiki pages, e.g. the *forwardlinks* denote the role HomerSimpson plays on other pages. Notice also on the sidebar that BartSimpson and others got listed under *Also SomeOne as InstanceOf* since those concepts share the same type – analogously *Also TheSimpsons as PartOf,* which lists those concepts that share the same PartOf value, namely TheSimpsons.

## 2 Related Work

It may seem that the wiki way and the semantic way diverge and resemble two different worlds [5]. Wikis oppose full fledged ontology editors, such as Protégé [6], Swoop [7], or pOWL [8]. To bridge the gap between the unstructured and structured world various ideas exist. The Mangrove project [4], [9] e.g. aims to overcome the obstacles of semantic markup by defining a small set of elements that are to be used to tag data such as people's names, contact information, and publications. The Mangrove search engine "understands" the extended markup to support semantic queries.

Authoring unstructured information is straightforward being the natural way of content creation. Wikis offer exactly this way of content creation. How can this experience be further enhanced? The following two approaches differ in the actual benefit for the wiki.

- **Support collecting general metadata about concepts portrayed by wiki pages using RDF.** RDF in wikis is of minor use for the wiki *per se* – it is more or less an easy way of creating RDF triples used or harvested by other Semantic Web applications.

- **Devise proprietary means of laying structure over wiki pages.** Adding structure to wikis by proprietary means immediately changes the wiki experience, but is of no further use for other Semantic applications.

For each approach some exemplary systems will be looked at next.

## 2.1 RDF within Wikis

Generally, one wiki page represents one concept with its RDF subject set to the page name. Alternatively, marked sections of wiki pages could each denote a different concept, i.e. one page could contain multiple subjects.

The PlatypusWiki [10] adheres to the first option. It resembles a promising effort to implement a general RDF wiki engine that supports the use of RDF vocabularies including Web Ontology Language (OWL) to represent metadata and relations between wiki pages. Content and metadata have to be edited on separate edit pages, though. Entering RDF triples is done via specific form fields for each subject, predicate, and object, and thus a somewhat irritating undertaking. Still, the engine does not (yet) enforce any constraints – inconsistencies are to be sorted out by the community.

Worth mentioning also is the OpenGuides.org wiki engine which fulfils the very specific purpose of a city guide listing possible spots of interests. Annotated with their geographical position the wiki supports querying for other places that are locally near a attractive spot. The according metadata is stored as RDF annotations that have to be again entered into various specific form fields on each wiki page.

## 2.2 Structure around Wikis

Wiki engines are a popular field of experimentation which results in an abundance of different implementations. Well known wiki webs and broadly used engines include: WikiWikiWeb[1], UseMod.com, MoinMoin[2], Wikipedia.org's MediaWiki, TWiki.org, JSPWiki.org, and ZWiki.org. Each has very individual feature sets. Common to all is the *backlinks* mechanism, which is used to get lists of pages linking to the current page.

1 Portland Pattern Repository Site and Software <http://c2.com/cgi/wiki?WikiWikiWeb>

2 MoinMoin is a Python Clone of WikiWiki. <http://sourceforge.net/projects/moin>

# 3 The SHAWN prototype

## 3.1 Usability in both Content Creation and Navigation

The ease of putting content online on the Web culminates in wikis, where the content of each page can be edited on the page itself. Adapting this approach the SHAWN prototype allows both entering structural information or metadata and further free text describing the concept – all within the same edit box on the wiki page. No going back and forth between content and metadata view. With typical field value pairs one creates RDF-like triples (subject-predicate-object) with subject being the page (name) itself, property the field, and object/literal the assigned value.

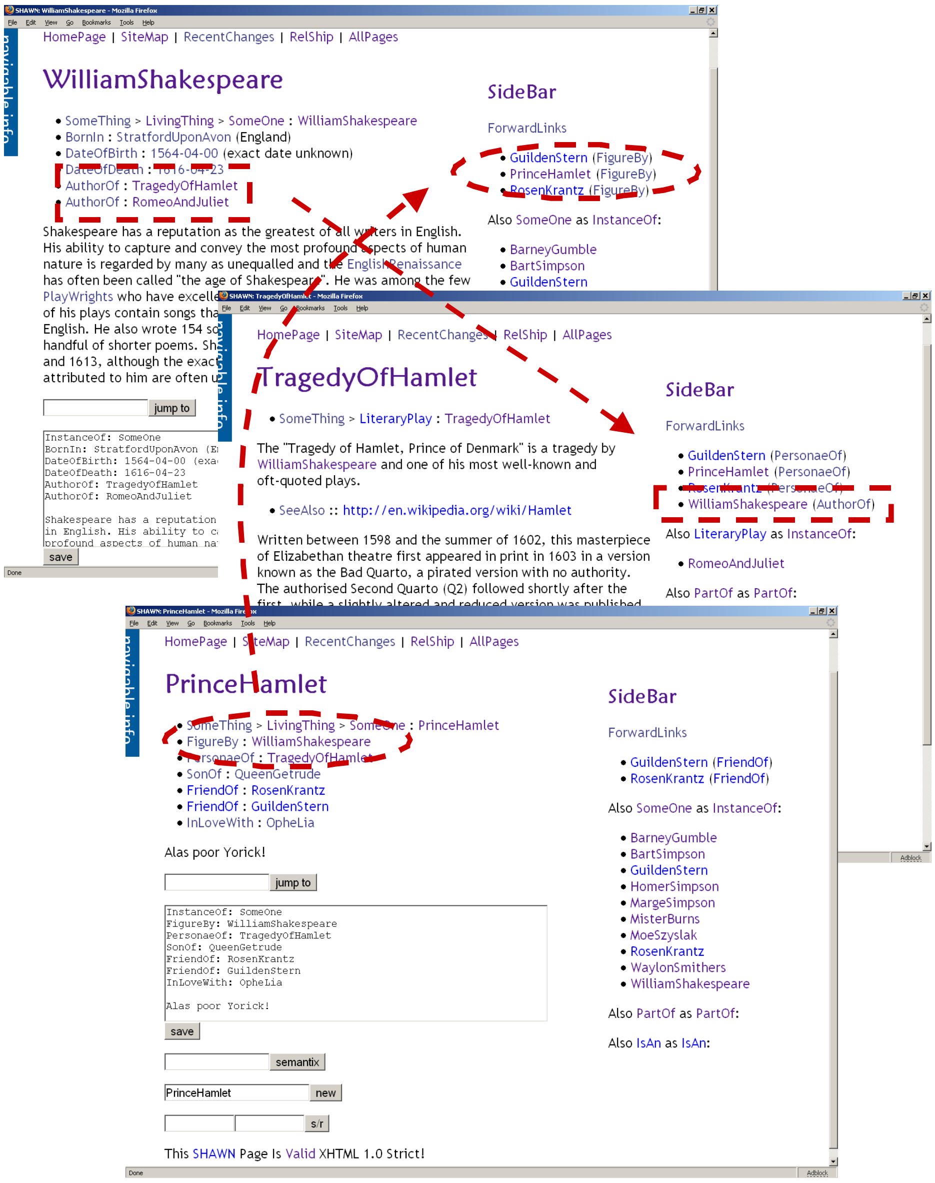
The structural information is instantly used to enhance navigation. Navigational aids as demanded by J. Nielson [11] for maximum usability of websites include answers to the questions “Where am I” and “Where can I go”. The first gets credited within SHAWN with so called *breadcrumbs* that show the path to the current page from the root of the site or concept (following specific transitive relationship types). The second is resembled by what will be called *forwardlinks.* These display the page names of those pages that link to the current page via arbitrary relationship types entered within their concept wiki page. Thus, instant gratification shows the user the value of structuring the data.

## 3.2 Instant Gratification

Every concept, be it a subject or predicate in the sense of RDF, is resembled as wiki page. The *forwardlinks* show all to the current concept related pages. If a concept is used as predicate, i.e. when a wiki page takes the role of a relationship type, the wiki engine will list all triples that use this predicate. With this mechanism the user immediately sees all the other concepts that share the same property, showing also the objects or literal values used therein. For example, a model containing some persons as concepts involving a relationship type “LivesIn” would, on this page, show all other persons that have this property together with the actual values of these properties, i.e. the whole triples.

Due to the effortless approach of adding arbitrary facts to a concept (by merely entering field value pairs in the edit box of the wiki page), the user can add anything that comes into her mind regarding the concept and can make up any kind of field value pairs (i.e. relationship types or predicates) that seem reasonable at this moment. At a later stage, these could further be typed using an is-a relationship type to build hierarchies of relationship types. This will reduce the potential structural disorder of early creative sessions and an ontology will emerge by itself.

Instantaneously gratifying the user while structuring her content conflicts with enforcing possible integrity constraints. These have to be deferred to a later stage, e.g. when the exported OWL ontology ought to be used within ontology applications of stricter manners. McBride [12] points out that it has to be accepted that flawed information models will be prevalent on the Semantic Web, nevertheless.


SHAWN: WilliamShakespeare - Mozilla Firefox

HomePage | SiteMap | RecentChanges | RelShip | AllPages

WilliamShakespeare

- SomeThing > LivingThing > SomeOne : WilliamShakespeare
- BornIn : StratfordUponAvon (England)
- DateOfBirth : 1564-04-00 (exact date unknown)
- DateOfDeath : 1616-04-23
- AuthorOf : TragedyOfHamlet
- AuthorOf : RomeoAndJuliet

Shakespeare has a reputation as the greatest of all writers in English. His ability to capture and convey the most profound aspects of human nature is regarded by many as unequalled and the EnglishRenaissance has often been called "the age of Shakespeare". He was among the few PlayWrights who have excelle
of his plays contain songs tha
English. He also wrote 154 so
handful of shorter poems. Sh
and 1613, although the exact
attributed to him are often u

```
InstanceOf: SomeOne
BornIn: StratfordUponAvon (E
DateOfBirth: 1564-04-00 (exa
DateOfDeath: 1616-04-23
AuthorOf: TragedyOfHamlet
AuthorOf: RomeoAndJuliet

Shakespeare has a reputation
in English. His ability to c
profound aspects of human na
```

SideBar

ForwardLinks

- GuildenStern (FigureBy)
- PrinceHamlet (FigureBy)
- RosenKrantz (FigureBy)

Also SomeOne as InstanceOf:

- BarneyGumble
- BartSimpson
- GuildenStern

SHAWN: TragedyOfHamlet - Mozilla Firefox

HomePage | SiteMap | RecentChanges | RelShip | AllPages

TragedyOfHamlet

- SomeThing > LiteraryPlay : TragedyOfHamlet

The "Tragedy of Hamlet, Prince of Denmark" is a tragedy by WilliamShakespeare and one of his most well-known and oft-quoted plays.

- SeeAlso :: http://en.wikipedia.org/wiki/Hamlet

Written between 1598 and the summer of 1602, this masterpiece of Elizabethan theatre first appeared in print in 1603 in a version known as the Bad Quarto, a pirated version with no authority. The authorised Second Quarto (Q2) followed shortly after the

SideBar

ForwardLinks

- GuildenStern (PersonaeOf)
- PrinceHamlet (PersonaeOf)
- RosenKrantz (PersonaeOf)
- WilliamShakespeare (AuthorOf)

Also LiteraryPlay as InstanceOf:

- RomeoAndJuliet

Also PartOf as PartOf:

SHAWN: PrinceHamlet - Mozilla Firefox

HomePage | SiteMap | RecentChanges | RelShip | AllPages

PrinceHamlet

- SomeThing > LivingThing > SomeOne : PrinceHamlet
- FigureBy : WilliamShakespeare
- PersonaeOf : TragedyOfHamlet
- SonOf : QueenGetrude
- FriendOf : RosenKrantz
- FriendOf : GuildenStern
- InLoveWith : OpheLia

Alas poor Yorick!

```
InstanceOf: SomeOne
FigureBy: WilliamShakespeare
PersonaeOf: TragedyOfHamlet
SonOf: QueenGetrude
FriendOf: RosenKrantz
FriendOf: GuildenStern
InLoveWith: OpheLia

Alas poor Yorick!
```

PrinceHamlet

This SHAWN Page Is Valid XHTML 1.0 Strict!

SideBar

ForwardLinks

- GuildenStern (FriendOf)
- RosenKrantz (FriendOf)

Also SomeOne as InstanceOf:

- BarneyGumble
- BartSimpson
- GuildenStern
- HomerSimpson
- MargeSimpson
- MisterBurns
- MoeSzyslak
- RosenKrantz
- WaylonSmithers
- WilliamShakespeare

Also PartOf as PartOf:

Also IsAn as IsAn:

Figure 2: Screenshots of related wiki pages within the SHAWN prototype

The screenshots in Figure 2 depict three concepts around William Shakespeare’s tragedy Hamlet. Page *WilliamShakespeare* has a property *AuthorOf* set to *TragedyOfHamlet*. On this page, the *WilliamShakespeare* gets listed under the ForwardLinks in the SideBar, mentioning also the *AuthorOf* relationship type (highlighted box). Similarily, the *FigureBy* relationship type connecting *PrinceHamlet* and *WilliamShakespeare* (highlighted ellipse).

## 3.3 Flexibility of the Implementation

The prototype currently is implemented in about 500 lines of Perl, running on an Apache web server. Each wiki page is stored on the server as plain text file. After processing potential special wiki commands, the resulting text is transformed to XHTML using MarkDown [13]. Common *CamelCase* words act as wiki links and thus as general concepts; so called freetext wiki links (e.g. masking wiki links by enclosing with double square parenthesis) could be easily implemented as well. To enter the semantic metadata (property-value pairs) no special markup is needed. Field value pairs are typed into the wiki edit box simply as in `property: value`. These pairs get parsed from each page and interpreted as subject-predicate-object triples for further use in support of navigation at various spots on a wiki page.

**Navigational aids** are located at the top (fixed links such as HomePage, SiteMap, and RecentChanges – as defined in a special wiki page called GotoBar) and on the right-hand side (sidebar) where links dependent on the current page are displayed, especially the aforementioned *forwardlinks*. The sidebar also is a wiki page by its own (SideBar) and is to contain special wiki commands that get transformed to lists or trees of links to semantically related wiki pages. Those operators can be used anywhere on any wiki page. Placing them in the special sidebar page makes them being executed and the results displayed for each page accordingly. By default, all pages which contain the current page as object of any property get listed (including the property) as forwardlinks and e.g. all pages which are of the same type as the current page get listed in the sidebar, as well. Thus, *breadcrumbs* make it easy to go up in the hierarchy; forwardlinks open way to venture deeper into the site or into more specialized concepts without getting lost.

Only few properties are intrinsically known to the wiki engine for specific rendering of the wiki pages. These are at the moment "TypeOf" and "InstanceOf". The transitive type-of property and the instance-of property are used to render the breadcrumbs so that the user knows exactly where her page or concept is located in context. As usual, the breadcrumbs path gets concatenated via greater than symbols ('>'); instances get appended to the breadcrumbs via the colon notation (': instance'), common for instances.

As data backend various RDF relevant implementations will be scrutinized for its adoptability in this wiki context, especially the open source RDF database systems Sesame[3], RDF Schema Specific Data Base[4] (RSSDB), and TRIPLE [14] (building on XSB[5]). The whole page content itself may be put as one triple ([pagename] <hasContent> [content]). Further relationship data (field-value pairs) need to be extracted upon saving (creating/updating) a wiki page and then put into the triple store. Suggestions of how to store RDF in relational databases [15] were collected once by Sergey Melnik. Generally, concerning a pure relational backend the database schema would need to be very simplistic for wikis being maximally flexible data repositories.

[3] OpenRDF, home of Sesame <http://www.openrdf.org>
[4] The ICS-FORTH RFDSuite <http://athena.ics.forth.gr:9090/RDF/>
[5] Logic Programming and Deductive Database System <http://xsb.sourceforge.net>

### 3.4 Visualization and further Navigational Aids

As visualization of either trees of specific relationship types or whole graphs of the complete semantic wiki structure the prefuse toolkit [16] was chosen. This toolkit contains various layout routines to display hierarchical and graph data in an interactive fashion that allows changing focus of nodes in context. Integrated as Java applet into the wiki environment it further enhances navigation and offers a more complete overview of complex structures. Planned to implement are controls to change visibility and appearance (width and colour) of each relationship type edge to filter the visualization of large graphs.

## 4 Use Cases

The most prominent wikis – the original WikiWikiWeb by Ward Cunningham, and the Wikipedia – are **communities** that maintain pages about arbitrary concepts and about their users or contributors themselves. Suppose the users state on their pages such facts as what skills or interests they have, where they live, and whom they know personally on the wiki. With simple inference rules a user could determine e.g. those persons that she might ask for help in specific matters or just might want to meet to share common interests. Taking also a (transitive) “knows” property into account the result could be granulated down to those people the questioner knows personally or could get to know via other people.

Imagine a kind of **personal semantic notebook** with address book and social network of all the people you know, storing their contact data, the online conversations you had with them, as well as their publications. Now, suppose you would like to plan a trip or sabbatical where you wish to meet as many researchers as possible who share your interests. Combining properties or facts such as LivesIn, AuthorOf, CoversTopic, and InterestsIn could finally infer a list of people that match your research interests and at the same time all live relatively close together.

As last use case SHAWN may serve as **online learning experience** to help teachers and pupils alike in structuring and/or understanding complex pieces of information. In a literature class e.g. the teacher might ask the students to model the relationships between the characters in a Shakespearean play. – *“Who killed Polonius?”*

## 5 Discussion and Future Integration

The wiki approach is a very flexible way of structuring pieces of information. For a start, this entices feeding lots of data into the wiki. Later on, the pieces can easily be rearranged by merely changing some field value-pairs or doing site wide search-and-replaces. An important aspect in ontology creation within the Semantic Web is to adhere to given structures and use vocabularies already available. When referring to a concept that is already defined by some ontology or vocabulary it should be used or refined instead of re-invented to keep heterogeneity and integration efforts low.

The SHAWN prototype supports RDF vocabularies by merely stating equality of concepts with concepts defined in external resources, i.e. URIs. For the RDF/OWL export, a special relationship type "SameAs" may be used to denote that e.g. a concept "SomeOne" denotes the same concept as <http://xmlns.com/foaf/0.1/person>, and the export module will thus replace references to SomeOne with the URI from the FOAF vocabulary. The RDF generation will be run each time a page got edited to keep the "RDF behind" up-to-date and searchable for Semantic Web crawlers. Yet, the SHAWN wiki is not meant to be a full fledged RDF/OWL editor. The flexible approach of this prototype will not enforce any semantics.

### 5.1 Semantic Retrieval through Inferences

By exporting the emergent triples to RDF in its XML or N3 notation, inference engines in conjunction with some rules may deduce further triples. As inference processor Tim Berner-Lee's CWM [17] or Sean B. Palmer's EEP[6] are easily deployed[7]. The challenge is the integration within the wiki environment – a question of designing a usable input forms. This interface ought to support typed queries, i.e. asking for concepts of specific type as in "Which literary plays are placed in Italy?", whereby also plays should be returned that do not explicitly state Italy, but maybe Verona (assuming also that on the wiki page denoting Verona there's a transitive property LiesIn stating the region and the region finally stating the state, Italy).

Additionally, queries will support a combination of both semantic search as exemplified above and traditional text search based on information retrieval techniques such as term frequency/inverse document frequency (TF/IDF). Thus, the above query might return the play "Romeo and Juliet" also when the word "Italy" occurs in the free description of its concept. The other way round will be interesting to see as well: searching for some free text could propose a concept type the searched word resembles, e.g. by simply suggesting the one concept type of which the searched word most often occurs in its belonging instances.

## 6 Conclusion

The prototype of a semantic wiki application presented here under the name SHAWN shows how a exceptionally simplistic approach to structuring data could succeed. It is the ease of publishing content on the Web that needs to be pursued further on the Semantic Web. Already little effort ought to be credited immediately as it is the case in editing general wiki pages. Whether it is only correcting typos or adding valuable content to a wiki, the result is straight away visible to the contributor and the whole community.

[6] Eep3: CWM Clone and SW API <http://infomesh.net/2002/eep3>

[7] E.g. based on a wiki base containing transitive facts of by whom person A got introduced to person P1…Pn, and where these persons live, the inference processor incorporating two simple rules returned a list of places the person A could go to visiting her friends.

To entice lots of users onto new technologies instant gratification is of vital importance. With the straightforwardness of entering structural metadata to wiki pages in SHAWN the user gets instantaneously gratified by additional navigational links resembling the structure of the growing model. This may even elicit further ideas to be entered in the wiki. The underlying semantics of the emergent ontology offer the user all the possibilities for her data existing and yet-to-come Semantic Web technology has to offer.

## References

[1] Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American (2001)

[2] Dublin Core Metadata Initiative. Dublin Core Metadata Element Set. Online: http://dublincore.org/documents/dces

[3] Leuf, B., Cunningham, W.: The Wiki way : quick collaboration on the web. Addison-Wesley 2001, Boston, MA ; Harlow (2001)

[4] McDowell, L., Etzioni, O., Halevy, A., Levy, H., Gribble, S., Pentney, W., Verma, D., Vlasseva, S.: Mangrove: Enticing Ordinary People onto the Semantic Web via Instant Gratification. In: Proc. Second International Semantic Web Conference (ISWC 2003) (2003)

[5] Alon Y. Halevy, O. E., AnHai Doan, Zachary G. Ives, Jayant Madhaven, Luke McDowell, Igor Tatarinov: Crossing the Structure Chasm. Conf. on Innovative Database Research (2003)

[6] Knublauch, H., Fergerson, R. W., Noy, N. F., Musen, M. A.: The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications. In: Proc. International Semantic Web Conference (ISWC) (2004)

[7] Kalyanpur, A., Sirin, E., Parsia, B., Hendler, J.: Hypermedia Inspired Ontology Engineering Environment: SWOOP. In: Proc. International Semantic Web Conference (ISWC) (2004)

[8] Auer, S.: A Web Based Platform for Collaborative Ontology Management. In: Proc. International Semantic Web Conference (ISWC) (2004)

[9] McDowell, L., Etzioni, O., Gribble, S. D., Halevy, A., Levy, H., Pentney, W., Verma, D., Vlasseva, S.: Evolving the Semantic Web with Mangrove. University of Washington, (2003)

[10] Campanini, S. E., Castagna, P., Tazzoli, R.: Platypus Wiki: a Semantic Wiki Wiki Web. In: Proc. Semantic Web Applications and Perspectives (SWAP) - 1st Italian Semantic Web Workshop (2004)

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

[12] McBride, B.: Four Steps Towards the Widespread Adoption of a Semantic Web. In: Proc. International Semantic Web Conference (2002)

[13] Gruber, J. MarkDown. Online: http://daringfireball.net/projects/markdown

[14] Sintek, M., Decker, S.: TRIPLE - A Query, Inference, and Transformation Language for the Semantic Web. In: Proc. International Semantic Web Conference (ISWC) (2002)

[15] Melnik, S. Storing RDF in a relational database. Online: http://www-db.stanford.edu/~melnik/rdf/db.html

[16] Heer, J., Card, S. K., Landay, J. A.: prefuse: a toolkit for interactive information visualization. In: Proc. Submitted to User Interface and Software Technology (2004)

[17] Palmer, S. B. CWM - Closed World Machine. Online: http://infomesh.net/2001/cwm/