From Ontology to Wiki – Generating Cascadable Default Fresnel Style from Given Ontologies for Creating Semantic Wiki Interfaces

Lloyd Rutledge

Faculty of Informatics, Open Universiteit
Amsterdam, The Netherlands
Lloyd.Rutledge@ou.nl

Abstract. We describe a mapping from any ontology to Fresnel style data and from Fresnel data to form-based semantic wikis. This enables quick start-up of distributed data-sharing systems for given knowledge domains. Concepts from Model-Driven Development apply to make an extendible default data query, browse and annotate system derived from a given data model. Our technique automatically generates from any given ontologies Fresnel lenses triples that define a default target interface for data using those ontologies. Hand-written Fresnel can then cascade over this default interface style to let the user fine-tune it, similar to the layered approach in CSS. Our approach also processes the source ontologies and the default and fine-tuning Fresnel to generate an assistive browse-and-annotate wiki interface. The result is the quick initialization and facilitated maintenance of distributed and accessible interfaces for collaborative input of data on the Semantic Web.

Keywords: semantic wikis, annotation interfaces, semantic browsers, Fresnel, style sheets, RDFS, OWL

1 Introduction: Toward a Form-based Semantic Wikipedia

While the Semantic Web and wikis began as separate visions, various projects have been pulling them together in recent years. Their largest integration begins with Wikipedia[1], a widely used website for information entry and display. Its infoboxes[2] were originally used purely for displaying properties and values for given page topics in small tables along the right side of page displays. Then the DBpedia project converted Wikipedia page links and structured content, including that in infoboxes, into a large set of triples [1] that helped seed the Linked Data Cloud and has long served as one of its core sources for interlinking concepts. Wikipedia and the DBpedia project have shown how wikis can enable people to enter large amounts of linked data that gets widely used on the Semantic Web. Wikipedia page infobox content is and remains

[1] http://www.wikipedia.org/
[2] http://en.wikipedia.org/wiki/Help:Infobox

typed in manually as wiki template code, and its export as semantic data is designed and performed by a third party at a later phase. Fig. 1 shows an example of the code and resulting display for a Wikipedia infobox, along with some of the RDF triples that DBpedia exports from that page. This paper refers several times to Fig. 1 as Wikipedia interface functionality that this paper aims to enable wiki administrators to specify more efficiently.

The next step in bridging the Semantic Web-wiki gap came in the form of semantic wikis, which add data input and processing to the originally document-oriented wikis. The primary semantic wiki tool is Semantic MediaWiki [15]. Semantic MediaWiki is an extension of the wiki system MediaWiki, which is a widely available and extendable system, and on which Wikipedia runs. It supports making systems that have some equivalent functionality of the Semantic Web, such as data annotations and queries on them. In addition, Semantic MediaWiki wikis can export their data in Semantic Web form, thus enabling the posting of a wiki's data on the Linked Data Cloud.

Semantic Forms, an extension of Semantic MediaWiki, brings data and wikis even closer with Wikipedia infobox-like table displays that define how data derives from their presented contents [14]. In addition, Semantic Forms provides form-based user entry of the data that these displays present and that the semantic wiki can query for. This tool also provides an interface for creating these tables and forms. Semantics Forms is thus an interface-driven developer's tool.

Fresnel offers an ontology for specifying browsing interfaces for Semantic Web data, effectively providing triple-defined stylesheets for interacting with data [21]. Typical Fresnel-generated interfaces resemble Wikipedia infoboxes and Semantic Forms tables.

Tools that follow the Model-driven development approach in building data systems have the user start with a data model from which the system then generates a data browse-and-entry that developers can then modify further. This paper applies model-driven approaches to the Semantic Web and semantic wikis. In particular, model-driven development applies here to the generation of Semantic Forms code from the Semantic Web representation of given ontologies.

This work's goal is thus the automatic generation of semantic wiki annotation interfaces from given ontologies. Such automation enables quick start-up of distributed data-sharing systems for given knowledge domains. With this work's results, one should be able to take the encoding of an ontology and generate from it a wiki that helps users enter conforming data and then makes that data part of the Semantic Web.

After presented related work, this paper discusses the use of Wikipedia infobox style as a desired target for testing this approach. Then the paper describes the general architecture for and our implementation of this approach. This implementation includes a mapping from ontologies and Fresnel to constructs for specifying semantic wiki interfaces. Next comes this work's primary contribution: the automatic generation of default Fresnel style for given ontologies. What follows is a description of various ways how user can tailor this generated default style to efficiently specify the desired interface.

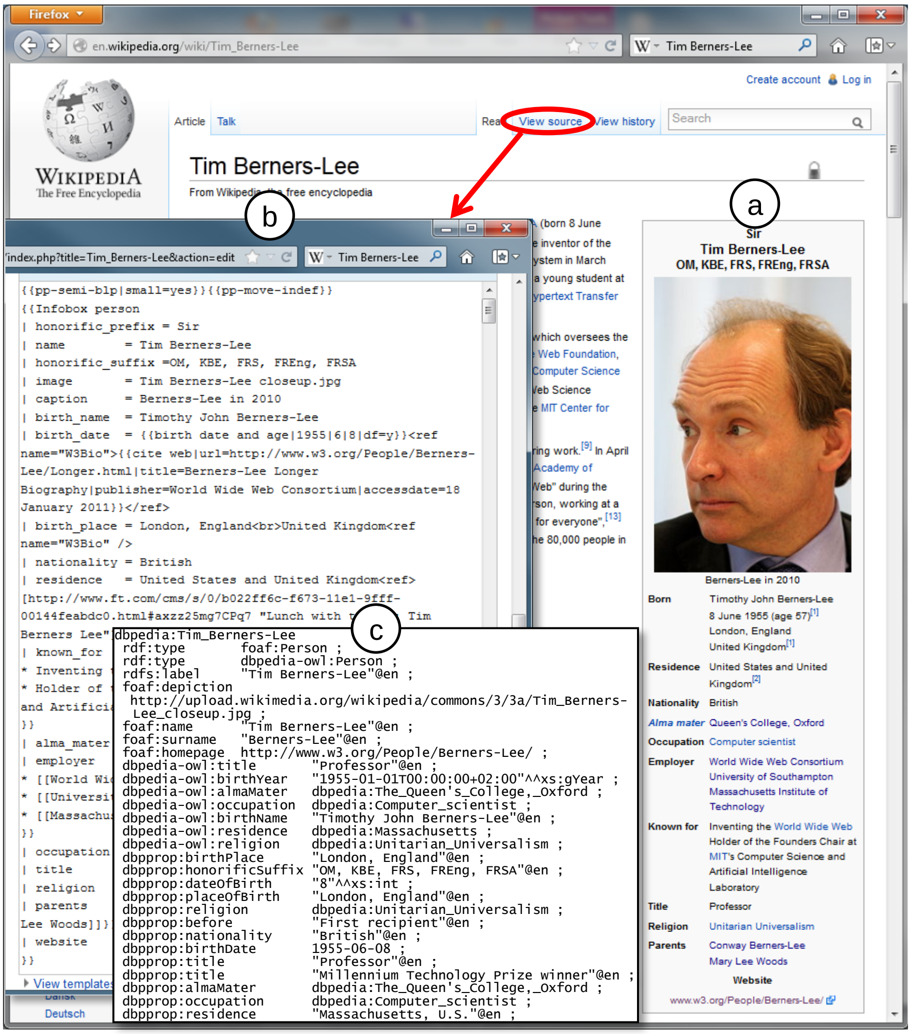

Fig. 1. Wikipedia's infobox for Tim Berners-Lee: (a) page display, (b) wiki infobox code and (c) a selection from its DBpedia export. (http://en.wikipedia.org/wiki/Tim_Berners-Lee)

2 Related Work: Data System Interface Development as Style

The separation of information from its user interface, which is the broader context of this work's goal, is an important practice in several areas of Computer Science. Cascading Style Sheets (CSS) established "style" as how browsers present World Wide Web documents. Meanwhile, the field of model-driven development developed parallels to CSS in the context of defining interfaces to data systems. For the combination of these types of systems, data on the web, Fresnel provides the specification of interfaces to the Semantic Web. Finally, Semantic Forms builds browse and annotation interfaces to data on semantic wikis. This section presents these various approaches to style for information systems, with the aim of unifying them in the rest of this paper.

2.1 Cascading Style Sheets

Cascading Style Sheets (CSS) define how XML documents should appear on web browsers [4]. They *cascade* in the sense that a specific style sheet can build upon the style defined in a more general style sheet. Appendix D of the CSS specification defines a *default* style sheet for HTML [4]. Thus, in the absence of any CSS for an HTML document, a browser renders the document as if applying this default CSS to it. Any CSS defined for an HTML document effectively cascades on this default style. This provides efficiency in defining style for HTML because a web developer only needs to specify the difference from the default style. Display according to the default style occurs in the absence any style code.

HTML is, of course, only one type of XML document set. Several browsers, such as Internet Explorer [12], assume default style sheets for all XML documents. These define how the browser displays an XML document in the absence of CSS, and what foundational style any provided CSS builds upon. We see the default style sheet for XML as a *domain-independent style* sheet and the default style sheet for HTML as the default for a specific domain of XML documents. This work applies these concepts to style for the Semantic Web by considering a default style for all Semantic Web data, default style sheets for ontology-defined domains that cascade on the domain-independent default, and the ability to define additional style that cascades on both of these defaults for efficiently tailoring of style.

2.2 Model-driven Development

Model-driven development tools handle data models as the starting point and core for developing information systems [17]. One such tool, Cathedron, generates a system interface for data display and input [16]. It can generate a default interface for an input database. We compare this to automatically generating a style sheet for a given domain, and apply this in our work here. In addition, Cathedron provides several means of specifying the interface beyond the default. One is the sorting of attributes as they appear in the display tables and input forms for given classes. Cathedron also offers overrides for different layers in the system and interface.

2.3 Semantic browsers and Semantic Style

Semantic browsers display triples for a given subject in a tabular form similar to that of infoboxes. However, semantic browsers typically have one type of table that applies to all properties and functions the same for all subjects. Tabulator is one example of a semantic browser [3]. Similarly, Semantic MediaWiki offers a “Browse properties” which shows all properties for a given wiki page in one table. We see such displays as domain-independent style, equivalent to how web browsers render XML documents without CSS.

Fresnel is a Semantic Web ontology for the presentation of data from given Semantic Web ontologies [21]. It thus effectively provides style sheets in the form of RDF triples. Fresnel encapsulates property assignment displays into lenses, which are typically associated with classes. Fresnel lenses are thus analogous with the class-based displays of Cathedron. Also like Cathedron, Fresnel enables sorting of properties rows in a display. Fresnel also provides referencing to CSS resources to define the style for presenting lens components. Such CSS thus overrides the default style that would apply. This use of CSS is also an example of how Fresnel, like Cathedron, provides default style override in different layers of the interface.

The */facet* browser in de ClioPatria framework process RDF-defined mappings between domain ontologies and an interface model to generate search-and-browse interfaces for those domains [11]. The /facet interface includes autocompletion to assist the user in entering terms that the underlying ontology recognizes. As potential next steps beyond these features in /facet, the authors include replacing their specific interface ontology with a broader one, naming Fresnel as a candidate. Our work build on these ideas from /facet by using RDF-defined mappings between ontology and interface, but also enabling cascading above a default mapping for each domain. We also use Fresnel as the target interface ontology to map to.

These systems and the approach this paper presents fall within established *categories* for ontology-enhanced user interfaces [19]. In this categorization, the domain of our approach is the *real world* but then any real world domain because it generates an interface from any given ontology. The complexity of our interface ontology is *medium*, since it uses Fresnel but also processes medium-level ontology constructs from source ontologies. The presentation form is *lists*. The interaction type is *view and edit* for *refining only*. The usage time is both *design and run time*.

2.4 Wikis

Semantic MediaWiki is well supported and under active development. It introduces a syntax for inclusion in wiki page code to annotate the concept a given page represents [15]. Data queries and formatted reports provide access to and presentation of this data. The tool is developed in a Semantic Web research context, but not all if its data process is strictly Semantic Web. However, it does offer an RDF export of the data is manages, including in the form of a directly connected SPARQL endpoint.

Semantic Forms extends Semantic MediaWiki with infobox-like data displays and input forms for populating them [14]. Like Cathedron and Fresnel, Semantic Forms provides grouping and sorting of properties into tabular displays. The fields and parameters Semantic Forms provides for defining the displays and input forms implies an interface model. Like /facet, Semantic Forms provides autocompletion for annotation via its forms. We explore here how to combine Fresnel with a Semantic Web definition of this interface model. Although Wikipedia uses neither Semantic Mediawiki nor Semantic Forms, it does apply MediaWiki to display data in tables and, via DBpedia, export that data to the Semantic Web as Fig. 1 illustrates.

Semantic Classes is a proposal for the creation of an extension to Semantic Forms offering a unified specification in XML for the various means of entering and presenting data for a given category [13]. Its proposed functionality is similar to the technique we present here. A key difference in our approach is the integration of existing web style technologies.

Early examples of semantic wiki's include SweetWiki [7], with a focus on folksonomic social tagging, and IkeWiki [23], with an early focus on direct Semantic Web compatibility. *Ontowiki* is alternative to MediaWiki and Semantic MediaWiki for Semantic Web functionality on a wiki interface [10]. It offers in-page data like Semantic MediaWiki and a form-based data entry interface comparable to that of Semantic Forms. While we use MediaWiki for our wiki environment, the work techniques presented here are equally applicable in other wiki systems such as OntoWiki. This paper builds upon these other wiki approaches by generating default Fresnel-defined interfaces for given ontologies, and allowing users to modify this style.

Wikidata is a project by Wikimedia that aims to provide a wiki for entering data in the context of other Wikimedia project such as Wikipedia [24]. As of this writing, Wikidata is being its second phase, which aims to generate Wikipedia infobox displays from Semantic Web data from Wikidata instead of human-written infobox template calls Wikipedia currently uses. Thus, Wikidata aims for the same kind of display from the same type of data source that this paper does. This paper's default and cascading style approach can thus apply to efficient specification of the appearance of Wikidata infoboxes.

2.5 Previous Work

In earlier work, we argue for "smart style" in the presentation of RDF by applying and adapting technologies for presenting XML documents on the World Wide Web. [18]. Later, we introduced OWL Wiki Forms (OWF) as courseware for Semantic Web courses [22]. This previous version of OWF lets students enter ontologies via the wiki and then see what browse and annotations interfaces they derive. We now extend OWF in this paper with the processing of preexisting external ontologies instead of user-entered ones, and by enabling user tailoring with Fresnel style.

3 Technique: Cascading Fresnel Style from Generated Defaults

We develop here a technique for processing Semantic Web ontologies to generate browsing and annotations interfaces. This technique is to map the components of a given ontology to Fresnel data for its display style, which in turn derives Semantic Forms special pages and template parameter values that define the interface for annotating and browsing instances from that ontology. Constructs in the formats RDF [2], RDFS [5] and OWL [20] serve here as the mapping input. Fig. 2 shows a diagram of the general process of OWL Wiki Forms. Table 1 shows OWF's mapping from ontological constructs to Semantic MediaWiki and Semantic Forms constructs, and which

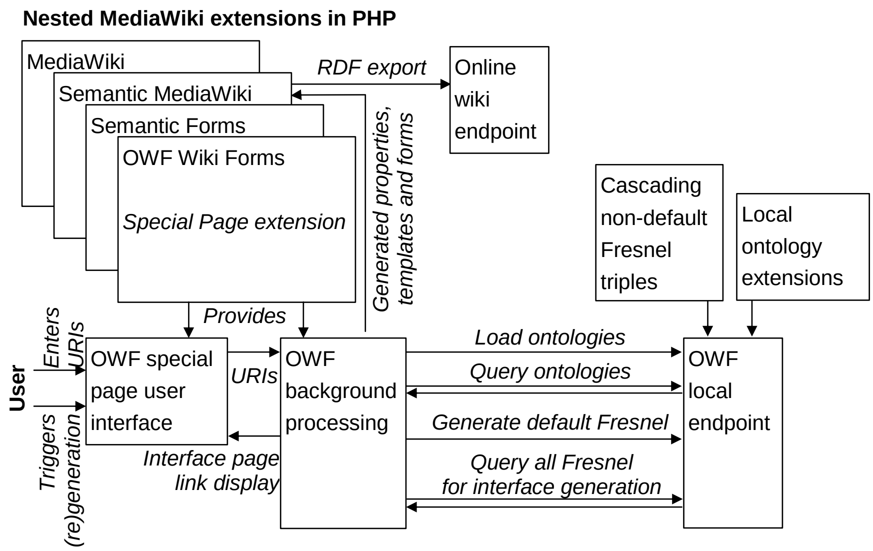

Fig. 2. Process diagram for OWF.

Fresnel constructs bridge them. The discussion that follows here refers to both this process diagram and table.

3.1 Test Sets: A Top Interface for a Top Ontology

The evaluation of the technique presented in this paper has three parts: an implementation, test input ontologies and a target interface. The implementation shows that this paper's technique can be built into a tool. It also provides a system to run tests on. Test input for the system comes is a widely used ontology. The target output for the system is a selected widely used interface whose data content corresponds with the test ontologies.

This work's evaluation technique is the comparison of an established handcrafted interface for data browsing and entry in a given domain with interfaces automatically generated from Semantic Web ontologies for that domain. Our source for established data browsing interfaces is Wikipedia's infoboxes. The selection criteria applied here for an infobox are that Wikipedia uses it frequently and that it corresponds with a frequently used ontology. Wikipedia's database report "Templates transcluded on the most pages" [3] shows how frequently each Wikipedia template, including infobox templates, is used. This report shows that the *person* infobox is among the three most frequently used infoboxes.

[3] http://en.wikipedia.org/wiki/Wikipedia:Database_reports/Templates_with_the_most_transclusions

Table 1. Construct mapping from ontologies via Fresnel to various wiki technologies.

Ontology			Fresnel	MediaWiki & extensions
URIs				[[EquivalentURI::]] [SMW]
				[[Imported from::]] [SMW]
rdfs:label			label	Pagename [MW] Label on form and box displays [OWF]
rdf:Property				Gets own Property: page [SMW]
rdfs:	Class		Lens defaultLens classLensDomain	Gets own Category: page [MW] [[Category:]] on topic page [MW] Gets own informbox (if domain) [OWF] Category [[Has default form::]] [SF]
	subclassOf			[[Category:]] on category page [MW]
	domain		showProperties	(Un)assign property to informbox [OWF]
	cascading only		hideProperties	
			showProperties =rds:seq	Sort properties in informbox
	range value is	class		autocomplete on category= [SF] property [[Has default form::]] [SF]
				[[Has type::Page]] [SMW]
		Literal some xsd:'s		[[Has type::String]] [SMW]
		other xsd:'s		[[Has type::Number]] [SMW]
		xsd:date(time)		[[Has type::Date]] [SMW]
		xsd:Boolean		[[Has type::Boolean]] [SMW]
				values=Yes,No | input type=radiobutton [SF]
		xsd:URI		[[Has type::URL]] [SMW]
	cascading only		image	
			externalLink	
			uri	[[Has type::String]] [SMW]
owl:	DataRange with oneOf			input type=Enumeration [SF] values= [SF] [[Allows value::]] [SMW]
	values	hasValue allValuesFrom		autocomplete on category=… [SF]
		someValuesFrom		
	cardinality	(min)...>0		mandatory [SF]
		(max)...=1		No #arraymap [SF], No list [SF]
		(max)...>1		Repeated fields [SF]
		(min)...>1		Repeated mandatory fields [SF]
		default		#arraymap [SF], list [SF]

Code and parameters generated by OWF from these technologies:
[MW] MediaWiki, [SMW] Semantic MediaWiki, [SF] Semantic Forms, [OWF] OWF-only
Planned or under development in OWF

The online services and prefix.cc analyzes how frequently namespaces are used on the Semantic Web [8]. It can thus indicate an ontology's popularity by the statistics for namespaces it is uses. Of the more than 4000 namespaces the site counts, the most popular namespace for a domain-specific ontology is FOAF[4] [6]. The FOAF ontology has important properties in common with those shown on the infobox for persons. We thus choose the person Wikipedia infobox and the FOAF ontology as representative test sets for testing the feasibility of this paper's approach. Future work can apply additional similarly prioritized infoboxes and corresponding ontologies to broaden this evaluation.

The existing interface serves as the target for what this technique generates in two ways. One is as the interface's general model and form, for which the technique is a mapping to for the processing of ontologies. The other source comes from the existing example of the interface, which serve as a desired interface for the given knowledge domain. Our technique should be able to generate an interface that is close to this example when processing ontologies from the same knowledge domain. A Wikipedia infobox applies here as the target browsing form. Our technique's aim is thus to generate browsing interfaces that are as close as possible to corresponding existing infobox. Given this benchmark ontology and interface, and a technique for using them, we show how OWF can process the former into the later.

3.2 Implementation: an Open MediaWiki Extension

Our implementation of this paper's approach is the latest version of OWL Wiki Forms (OWF). OWF is an open-source and available for download and installation[5]. As Fig. 2 illustrates, OWF is a MediaWiki extension built on top of Semantic MediaWiki and Semantic Forms. This latest version of OWF processes external OWL ontologies and Fresnel style triples to generate semantic wiki interfaces in the form of Semantic Forms page code.

The specific input to OWF is a list of ontology URI's. The specific output is the automatic creation wiki pages for categories, properties, templates and Semantic Forms form pages. Semantic MediaWiki provides much of OWF's internal wiki-based data processing. Generated Semantic Forms pages provide assistive form-based data entry. Part of this process is the generation and processing of Fresnel style triples for interface.

As a special page extension, OWF provides a MediaWiki "special page" with which the user can interact to specify and start the wiki interface generation. On this special page, the user enters the URIs of one or more ontologies to generate an interface from. The system then generates form-based data entry for the ontology on the wiki. The OWF special page then provides a table of links to all the wiki interface pages the system just generated. Fig. 3 shows an example of such an OWF special page display. Fig. 4 illustrates an example session with the generated interface, with a filled in informbox form, the resulting informbox code and the display it generates.

[4] `http://www.foaf-project.org/`
[5] `http://is.cs.ou.nl/OWF/`

Given the list of ontology URIs, OWF first generates Fresnel code that describes the structure of the target Semantic Forms-based interface. The wiki interface is then generated by querying this Fresnel data. This Fresnel data that OWF queries for includes the generated default interface. It also includes any additional applicable Fresnel data, which enables tailoring and extending the default style. OWF replaces Wikipedia's infoboxes with informboxes, which are MediaWiki templates that display page data on an infobox-like table and define the form with which the user enters that data.

This wiki-based implementation enables packaging interface specification created with OWF for distribution and installation on other wikis. What such a target wiki would need to start with is a fully installed OWF extension. Providing ontology URIs in the package enables OWF default generation. Additional ontology triples that only apply to that wiki can be put online at such a provided URI. Alternatively, an RDF file delivered with the package can include these triples, and the administrator can then provide a local URI for it that OWF can use. Similarly packaged URIs or RDF files can also include non-default Fresnel specifications for the interface. Finally, the distribution can include a MediaWiki XML import file containing any additional wiki template code for use in the generated informbox code.

3.3 Generating Default Fresnel Style from OWL Ontology Triples

The OWF approach is to generate a default style for the given ontologies in the form of Fresnel triples. This generation of default Fresnel triples can apply in principle to any system that processes Fresnel to set up semantic user interfaces. The OWF implementation generates this default Fresnel style and then processes it to create the semantic wiki components that make up the corresponding interface.

In Fresnel, a "lens" is one type of data display, equivalent to an infobox. The core aspects of the generation of default Fresnel style are creating Fresnel lenses for classes that need them and assigning properties to each lens with the showProperties property. OWF processes property domains to do so. It first makes a lens for each class that is a domain for a property. This includes the class owl:Thing, thus also creating an informbox for properties with no explicit domain. OWF then assigns properties to the lenses for the classes that are their domains.

Our OWF implementation maximizes the use of SPARQL to generate default Fresnel from source ontologies. Using SPARQL offers an implementation-independent definition of the default mapping. This makes the mapping more easily readable by human system developers. In addition, other interface generation systems can integrate and process it than the predominately PHP code of the OWF implementation. OWF uses SPARQL 1.1 Update [9] to put the generate triples directly in the triple store that the implementation later queries to generate the wiki.

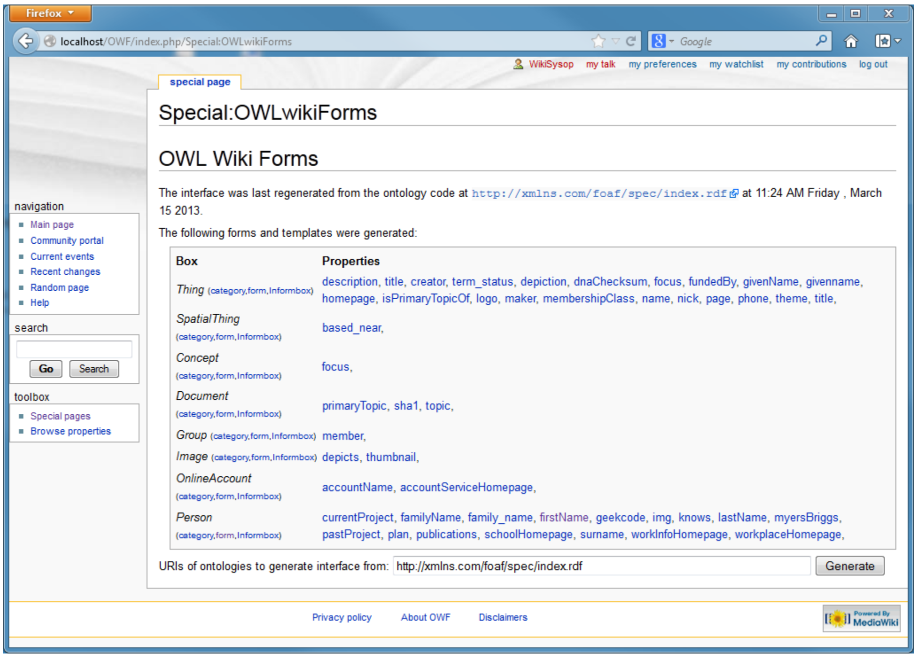

Fig. 3. Report displayed after OWF generates the default interface for the FOAF ontology

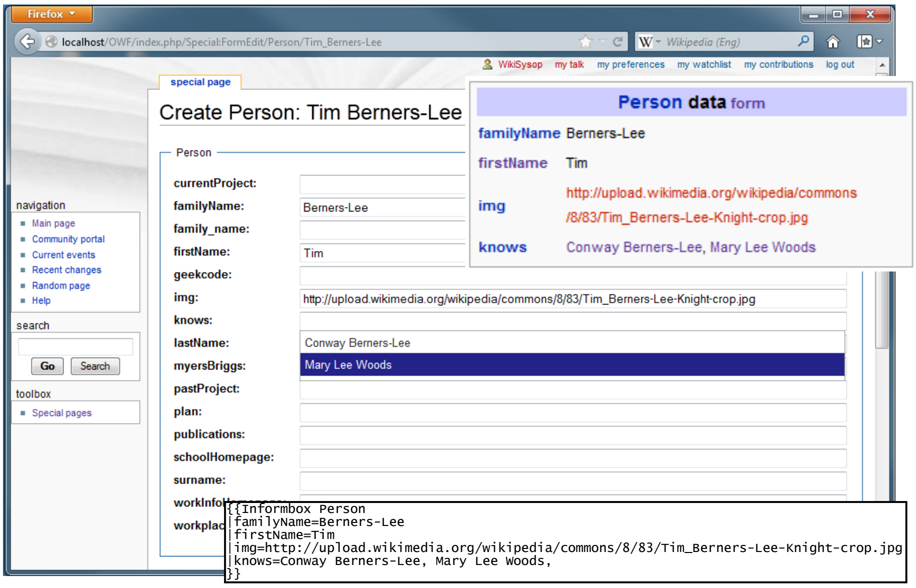

Fig. 4. Example of filling in a Person form from the default OWF interface for the FOAF ontology, with autocompletion for the property field “knows”, the generated informbox code, and the resulting informbox display.

The OWF implementation processes Fresnel lenses to generate corresponding informbox displays along with their corresponding forms. Its processing of the source ontologies also generates the following wiki components: MediaWiki categories and templates, Semantic MediaWiki property pages and property assignments in other generated wiki pages, and Semantic Forms form pages. Processing property data types determines how their values appear on page displays and how the user enters them on a form. Semantic Forms autocompletion parameters are derived from property ranges.

Given a generated default style for a given set of ontologies, there are various ways of cascading style on top of it for defining semantic wiki interfaces. These means of cascading style are discussed in the following subsections.

3.4 Cascading Ontologies

One means of cascading style with the OWF approach is by cascading ontologies. Here, a cascading ontology is simply an ontology that builds upon other ontologies by relating its resources with those of others. This can apply to this paper's example by adding the properties that appear in the Wikipedia person infobox but do not appear in the input ontologies. Making a new ontology with those properties and making their domain be foaf:Person will cause the generated default to include those properties in the generated Fresnel lens for foaf:Person. The URI for this new ontology can be included in the list of ontology URIs entered on the OWF special page.

3.5 Cascading Fresnel

This approach cascades ontologies simply by adding new triples that extend the data model. It can also cascade Fresnel style beyond the generated domain-specific default simply because it processes Fresnel triples regardless of their origin. That is, OWF processes both the generated default Fresnel triples and any relevant Fresnel triples entered in the triple store by a human user. This is similar to how HTML browsers process referenced CSS stylesheets that website administrators make, which cascade on the browsers default CSS stylesheet for HTML. When generating the wiki interface, a system can query for the new Fresnel triples from the same endpoint as the generated default Fresnel triples, effectively making no distinction between them.

Such cascading Fresnel style applies in this paper's example with fresnel:hideProperties triples that block inclusion of certain FOAF properties in the wiki interface, generating the form display in Fig. 5. The fresnel:showProperties can also apply to sort the properties that remain on the informbox to have the same order as the target infobox by making its value be an rdf:list ordering those properties.

Fig. 6 shows another step in overriding default rendering of FOAF to emulate the target presentation from Wikipedia. Here, the informbox displays an image as an image instead of as a URI. To achieve this, a Fresnel triple is added to the endpoint OWF uses that assigns fresnel:image to the fresnel:value property for formatting foaf:image property values.

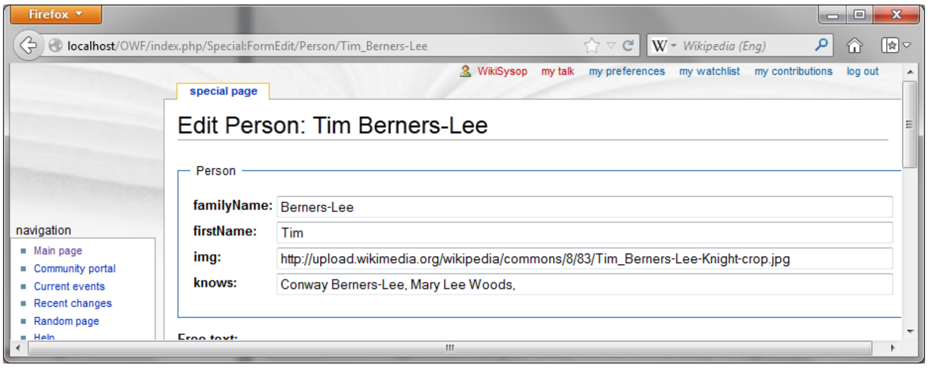

Fig. 5. FOAF person form resulting from cascading Fresnel style hiding undesired properties.

Our OWF implementation currently only supports a subset of Fresnel constructs in its querying for Fresnel style for generating wiki interfaces. However, this approach of cascading ontologies as input to generating default Fresnel style that can combine with beyond-default cascading Fresnel style can be applied to making interfaces on any Fresnel implementation. Furthermore, the OWF implementation can be used to input the external and cascading ontologies to generate the default Fresnel code than can then be exported for use in other Fresnel implementations.

3.6 Cascading CSS

While Fresnel introduces constructs for defining the general structure of semantic data interfaces, it refers to CSS code for defining the more detailed aspect of page rendering that CSS defines. This this approach lets one add Fresnel triples that refer to CSS resources. When processed, such Fresnel-linked CSS will cascade over any CSS that is otherwise linked to the generation of the browser display. OWF-generated systems can thus cascade CSS stylesheets via Fresnel.

3.7 Cascading Wiki Templates

There are of course aspects of semantic wiki interfaces that neither ontologies nor Fresnel nor CSS can define, and thus that only semantic wiki page code can specify. The OWF extension allows developers to effective cascade wiki code on top of the wiki code generated by processing the default style specification and any cascading ontologies, Fresnel and CSS. Each generated informbox template checks for the existence of a template on the wiki named “InformboxTop” followed by the informbox’s name. If a wiki developer had created such a template then it gets transcluded just after the informbox on the rendered wiki page display. An example use case for such cascading wiki templates is the addition of a specialized wiki-generated display on any page for a given type of informbox.

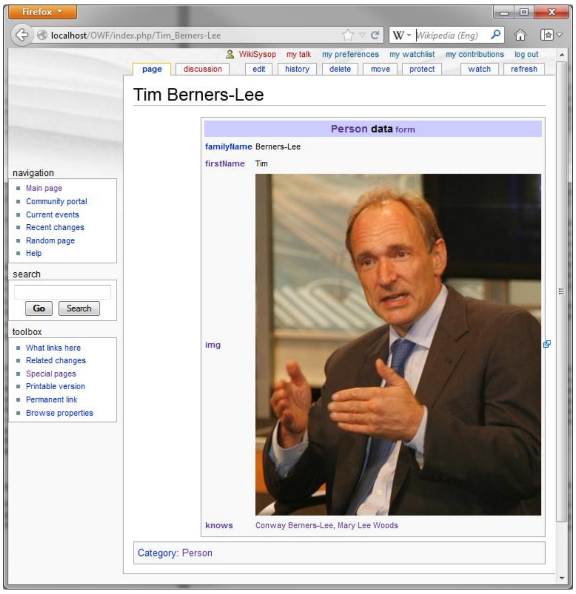

Fig. 6. Informbox display from cascading Fresnel style formatting foaf:img as an image.

4 Conclusion: Further with Cascading Semantic Style

This paper described applying default and cascading style to generate annotate-and-browse interfaces from given ontologies. An extension of MediaWiki, the implementation is readily installed and made accessible online to distributed users. Fresnel is applied to define the interface style. Adoption and adaptation of the CSS approach enable cascading style definitions on top of such default interfaces. This cascading occurs in terms of local extensions to the given ontologies and as tailored Fresnel code that extends the generated default semantic style. This approach is evaluated by replicating a Wikipedia infobox display. While Wikipedia infoboxes serve here as proven interfaces to aim for, this approach can hypothetically implement Wikipedia infoboxes themselves. This approach enables developers to efficiently make and maintain distributed browse and annotate interfaces for given ontologies.

References

1. S. Auer, C. Bizer, G. Kobilariov, J. Lehmann and Z. Ives. "DBpedia: A nucleus for a web of open data", in *The 6th Int'l Semantic Web Conference (ISWC 2007)*, Busan, Korea, 2007.
2. D. Becket (eds.). *RDF/XML Syntax Specification (Revised)*, W3C Recommendation, 2004.
3. T. Berners-Lee, Y. Chen, L. Chilton, D. Connolly, R. Dhanaraj, J. Hollenbach, A. Lerer, and D. Sheets. "Tabulator: Exploring and analyzing linked data on the semantic web". In *Proceedings of the 3rd Int'l Semantic Web User Interaction Workshop (SWUI 2006)*, 2006.

4. B. Bos, T. Çelik, I. Hickson, H. Wium Lie. *Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification.* W3C Recommendation, 2011
5. D. Brickley and R.V. Guha (eds.). *RDF Vocabulary Description Language 1.0: RDF Schema*, W3C Recommendation, 2004.
6. D. Brickley and L. Miller (eds.). *FOAF Vocabulary Specification 0.98*. 2010.
7. M. Buffa and F. Gandon. 2006. SweetWiki: semantic web enabled technologies in Wiki. In *Proceedings of the 2006 Symposium on Wikis (WikiSym '06)*, Odense, Denmark, August 2006.
8. Digital Enterprise Research Institute (DERI). *prefix.cc namespace lookup for RDF developers*, Accessed 04/03/2013, 2013. http://prefix.cc/
9. P. Gearon, A. Passant, A. Polleres. *SPARQL 1.1 Update.* W3C Proposed Recommendation 08 November 2012, 2008.
10. N. Heino, S. Dietsold, M. Martin and S. Auer. "Developing semantic web applications with the ontowiki framework", *Networked Knowledge - Networked Media, Studies in Computational Intelligence*, vol. 221, no. 17, pp. 61-77, 2009.
11. M. Hildebrand and J. van Ossenbruggen. "Configuring Semantic Web Interfaces by Data Mapping",.In *Workshop on Visual Interfaces to the Social and the Semantic Web (VISSW2009).* 2009, February.
12. S. Howlett and J. Dunmal. "Beyond ASP: XML and XSL-based Solutions Simplify Your Data Presentation Layer", *MSDN Magazine*, November 2000.
13. Y. Koren. Semantic Classes proposal. *Semantic MediaWiki Plus - SMW+ A Semantic Web Enterprise Wiki*, Semantic Wiki Discussion Session 9, 2010.
14. Y. Koren. *Semantic Forms*, Accessed 04/03/2013, 2013. http://www.mediawiki.org/wiki/Extension:Semantic_Forms.
15. M. Krötzch, D. Vrandecic, M. Völkel, H. Haller and R. Studer. "Semantic wikipedia," *Journal of Web Semantics,* no. 5, p. 251–261, September 2007.
16. Mattic B.V., *Cathedron Manual for the Preview / Field Test Release*, 14 September 2007
17. S. J. Mellor, T. Clark and T. Futagami. "Model-driven development: guest editors' introduction". *IEEE software*, 20(5), 14-18, 2003
18. J. van Ossenbruggen, L. Hardman, and L. Rutledge. "Combining RDF Semantics with XML Document Transformations". In: *Journal of Web Engineering and Technology (IJWET)* (volume 2, number 2/3, pages 248-263), December 2005.
19. H. Paulheim and F. Probst. "Ontology-Enhanced User Interfaces: A Survey." *Semantic-Enabled Advancements on the Web: Applications Across Industries*. IGI Global, 2012. 214-238. Feb. 2012. doi:10.4018/978-1-4666-0185-7.ch010
20. P.F. Patel-Schneider, P. Hayes and I. Horrocks (eds.). *OWL Web Ontology Language Semantics and Abstract Syntax*, W3C Recommendation, 2004.
21. E. Pietriga, C. Bizer, D. Karger and R. Lee. "Fresnel: A Browser-Independent Presentation Vocabulary for RDF," in *The 5th Semantic Web Conference*, Athens, Georgia, USA, 2006.
22. L. Rutledge and R. Oostenrijk. "Applying and Extending Semantic Wikis for Semantic Web Courses", In: *Proceedings of the 1st Workshop on eLearning Approaches for the Linked Data Age (Linked Learning 2011)*, Heraklion, Greece, May 2011.
23. S. Schaffert. "IkeWiki: A Semantic Wiki for Collaborative Knowledge Management", In: Proceedings of the 15th IEEE Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE 2006), Manchester, U.K., June 2006.
24. D. Vrandečić. "Wikidata: a new platform for collaborative data collection", Keynote presentation at *Semantic Web Collaborative Spaces Workshop (SWCS 2012)*, Lyon, France, April 2012.