Addressing the RDFa Publishing Bottleneck

Xi Bai
supervised by Prof. Dave Robertson
School of Informatics, University of Edinburgh
10 Crichton Street
Edinburgh EH8 9AB, UK
xi.bai@ed.ac.uk

ABSTRACT

In the more dynamic environments emerging from *ad hoc* and peer-to-peer networks, our research has explored the extent to which Web-based knowledge sharing as well as community formation require automation to understand human-readable content in a more distributed manner. RDFa is a syntactic format which can leverage this issue by allowing machine-readable data to be easily integrated into XHTML Web pages. Although there is a growing number of tools and techniques for generating and distilling RDFa, comparatively little work has been carried out on publishing existing RDF data sets as an XHTML+RDFa serialization. This paper proposes a generic approach to integrating RDF data into Web pages using the concept of automatically discovered "topic nodes". RDFa^2 is a proof-of-concept implementation of this approach and provides an on-line service assisting users in generating and personalizing pages with RDFa. We provide experimental results that support the viability of our approach to generating Web documents such as FOAF-based online profiles as well as RDF vocabularies with little user intervention.

Categories and Subject Descriptors

H.3.5 [**Information Storage and Retrieval**]: Online Information Services—*Data sharing*

General Terms

Algorithms, Design

Keywords

Semantic enhancement, federated markup, linked data, RDFa

1. INTRODUCTION

Many systems exist for community formation in extensions of traditional Web environments (e.g., social network sites) but little work has been done on forming and maintaining communities in the more dynamic environments emerging from *ad hoc* and peer-to-peer networks. Our research has explored how Web-based knowledge sharing for community formation can use automation to understand and digest human-readable content in a more distributed manner [3].

WWW 2011, March 28–April 1, 2011, Hyderabad, India.
ACM 978-1-4503-0637-9/11/03.

As part of the Semantic Web initiative to promote machine-readability of Web documents, RDFa (a W3C recommendation for more than two years) has been designed so that "authors can markup human-readable data with machine-readable indicators for browsers and other programs to interpret" [1]. An increasing number of tools for processing RDFa have been developed which leverage the existing raft of techniques for processing standard RDF. Hundreds of thousands of FOAF documents have been created (semi-)automatically or manually but due to the lack of readability of triples, many of these documents are hidden in repositories or behind SPARQL endpoints which are not accessable to users without expertise. Automatic information processing and integration require Web documents to be not only human-readable but also machine-readable. While RDFa makes it easy for Web authors to manually add small amounts of semantic markups to XHTML documents, RDFa also offers the potential to transform pre-existing machine-readable data into human-readable format. So far, this has received relatively little attention. We propose a generic approach to converting RDF documents[2] into XHTML documents. Due to space limitations, in this paper we summarize the results but more technical details are available online [3]. A key ingredient, which we will describe in more detail below, involves the identification of one or more "topic nodes" in the RDF context(s) to guide the injection of RDFa into an XHTML template.

A proof-of-concept of this approach has been implemented under the name RDFa^2, and has been available as an online service [4]. RDFa^2 runs within standard Web browsers, and allows users to customize its output in two ways: either by being guided to modify the generated raw data in an edit window (with on-the-fly preview) or by revising the generated XHTML template, which can be saved to local storage for future reuse.

The remainder of this paper is organized as follows. Section 2 describes the preprocessing of "RDF contexts" required by our approach. In Section 3, we propose a hybrid topic-node discovery method based on weighted occurrences of nodes as well as heuristic properties. Section 4 details how users can be assisted in creating and customizing Web content with RDFa based on our approach. Section 5 explores how RDFa^2-assisted data integration is compliant with

[1] http://www.w3.org/TR/xhtml-rdfa-primer/
[2] Here and in the rest of the paper, we take "RDF document" to subsume any documents containing RDF triples.
[3] http://www.inf.ed.ac.uk/publications/report/1391.html
[4] http://demos.inf.ed.ac.uk:8836/rdfasquare

Linked Data [5] principles. Section 6 evaluates the performance of our approach through case studies on republishing FOAF profiles and vocabularies as well as aggregating linked datasets. Section 7 reviews related work on processing RDFa, while Section 8 draws conclusions and indicates future work finally.

2. TOPIC NODES AND TOPIC TREES

Our RDF documents to XHTML+RDFa documents transformation algorithm is based on automatically generated templates. These templates are schematic XHTML documents, and have a tree structure. By contrast, the RDF data model is a graph, and cannot be converted to a single tree without duplicating re-entrant nodes. In order to overcome this problem, the conversion from RDF requires users to select a specific node in the RDF graph which then forms the root of a tree of RDF statements. Which node should the user choose? In practice, this seems to follow straightforwardly from the user's goals, namely to focus on the resource which is his or her of interest in the resulting XHTML page. For instance, in the case of a FOAF file, the obvious resource to choose is the value of the `maker` (or `primaryTopic`) FOAF property.

The node that is targeted in this way is called the *topic node*. The RDF document from which the topic node is derived is called the *RDF context*, and relative to a context $\mathcal{C}$, a set of RDF statements rooted in a topic node is called a *topic $\mathcal{C}$-tree*. We distinguish between two kinds of topic trees, depending on the semantic role of the topic node. Given a resource r, context $\mathcal{C}$, and RDF statement (s, p, o), the *subject (topic) $\mathcal{C}$-tree based on r* is defined as $\{(s, p, o) \in \mathcal{C} \mid s = r\}$, and similarly for the *object (topic) $\mathcal{C}$-tree based on r*.

The notion of a topic tree for a topic node is essentially the same as a *bounded description* of a resource; that is, where "a sub-graph can be extracted from a dataset which contains all of relevant properties and relationships associated with a resource" [6]. Note that a topic node is not necessarily the global topic of an RDF document; rather, it corresponds to a resource in the document which the user regards as interesting enough to represent in XHTML.

Figure 1 illustrates the selection of a subject topic tree from an RDF context. For the sake of brevity, we have omitted the name spaces (henceforth abbreviated as NS) of all properties here. In this figure, circles denote resources and squares denote literals. The dark gray node is the current topic node while the sub-graph surrounded by the dashed line is the subject topic tree for this node. The labeling information about the resources in the subject position are also included in the topic tree in order to make the resources themselves human-readable on the RDFa-embedded page.

Although the most straightforward use case for our approach creates a standalone XHTML page from an RDF document, we also want to accommodate cases where the output of the system is inserted as a snippet into a larger XHTML document. Taking Sir Tim Berners-Lee's FOAF document and homepage as an example, `RDFa`^{2} generated an RDFa snippet that is ready to be inserted into the `<body>` section of the homepage as illustrated in Figure 2.

[5]http://www.w3.org/DesignIssues/LinkedData.html
[6]http://patterns.dataincubator.org/book/

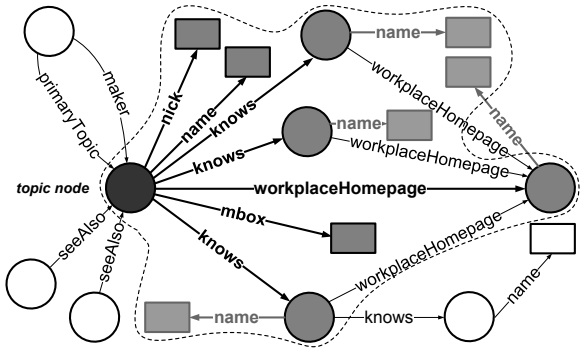


Figure 1: *Subject (topic) $\mathcal{C}$-tree* of a FOAF document

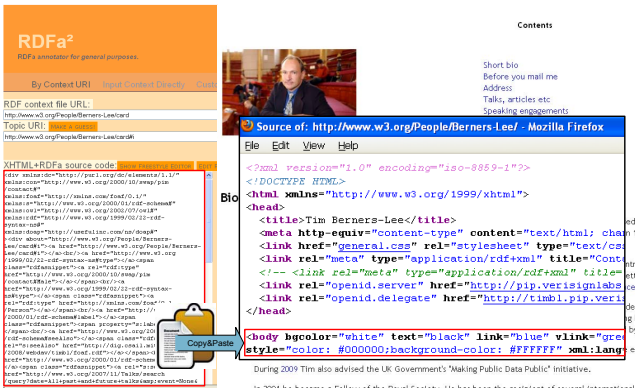


Figure 2: Inserting RDFa into an existing Web page

3. TOPIC NODE DISCOVERY

In the preceding section, we assumed that topic nodes will be selected by the user. However, this requires the user to understand the basic syntax of the RDF context inside which these node are represented. In order to overcome this barrier to usability, our approach also has the ability to automatically discover a candidate set of topic nodes which can be offered to the user thereafter. In this section, we discuss methods for discovering and listing topic nodes.

3.1 Occurrence-Based Discovery

One way of finding topic nodes from the given RDF context is to query this document for URIs with properties that are diagnostic of topic-hood, such as aforementioned `foaf:primaryTopic` or `foaf:maker` in FOAF files. However, not all RDF documents contain such properties, and even in FOAF files which do employ them, they do not always take semantically appropriate values. Consequently, topic nodes cannot reliably be detected just in terms of the semantics of statements in the RDF context itself. Xiang et al. compared five measurements used for automatically summarizing ontologies and the experiments showed that the weighted in-degree centrality measures have the best performance [7]. However, as analyzed in [2], for the case that the target RDF documents not only contain ontologies but also contain a large number of RDF individuals, this measurement usually does not work effectively. Moreover, according to the definition of the inverse property, each property can have its own inverse property. Therefore, for each RDF node, its in-degree and our-degree should be equivalently im-

portant. In this paper, we propose an improved algorithm for semi-automatically discovering and recommending topic nodes. Since the RDF data model is a directed graph and nodes are connected to one another through directed edges, one solution for discovering the topic node is based on node connectivity. In other words, the more edges (outgoing or incoming) a node has, the more important it is likely to be. In order to maximize the accuracy of this heuristic, our algorithm selects the top n most highly connected URIs and offers them to users for subsequent confirmation.[7] Perhaps not surprisingly, this algorithm works especially well for RDF documents such as FOAF files that usually do have a central topic.

3.2 Dealing With Multiple Topic Nodes

We do not want to exclude the possibility of the user selecting more than one topic node from a given RDF context. For instance, a user may wish to render the FOAF document vocabulary (i.e., encoded as a set of RDF statements) as XHTML, and in this use case, all of the nodes `foaf:Person`, `foaf:Agent` and `foaf:Document`, for instance, should be treated as topics. We can use multiple templates to make the user achieve this goal. As mentioned before, a user is normally focused on a single object being integrated and temporarily ignores other objects. Once a user selects a temporary topic node, a map tree, a template as well as an XHTML+RDFa page, will be generated based on node occurrences. Our approach allows users to select multiple topic nodes and for each topic, a map tree and a template will be generated. Meanwhile, the relevant NSs are also grouped and displayed on the final page. Thereafter, the generated XHTML+RDFa snippets will be automatically combined into a single snippet. Following Figure 2, the screenshot in Figure 3 illustrates multiple topic nodes derived from a single RDF context (a FOAF document) were discovered and recommended to users by RDFa^2.

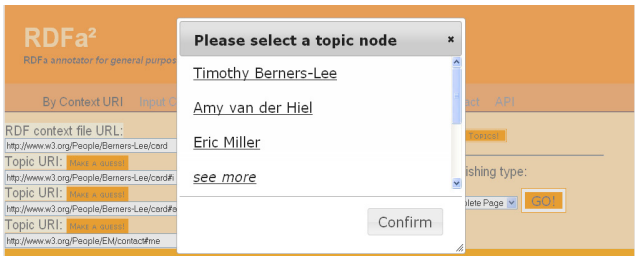


Figure 3: Screenshot for discovering multiple nodes from a single RDF context

4. FEDERATED INTEGRATION AND CUSTOMIZATION

It is common for users to publish an XHTML+RDFa page using triples from different RDF sources (or in our terminology, from different contexts). We can accommodate this in a way similar to our approach to dealing with multiple topic nodes. Our approach supports federated integration by managing the NSs derived from different RDF documents separately and combining them at the final stage. However, it should be noted that different vocabularies do not necessarily employ the same QName prefix for a given NS. *Prefix.cc* (*PCC*) [8] alleviates the issue that RDF documents involve different prefixes indicating the same NS or the same prefix indicating more than one NS by allowing users to look up the collected NSs on *PCC* and vote for their favorite ones. Nevertheless, it is difficult if not impossible to stop people from using ambiguous prefixes. Our approach can automatically detect if a prefix is ambiguous across a set of contexts, and will synthesize new prefixes to ensure disambiguation. Figure 4 illustrates how RDFa^2 assists users in creating Web pages annotated with RDF triples derived from different data sources. Users inform RDFa^2 of the target one or more RDF contexts by selecting one or more URLs and RDFa^2 then retrieves these documents on the fly, each of which forms an RDF context. After the topic nodes are selected, triples related to them will be extracted. Finally, the page with RDFa annotation will be sent back to users.

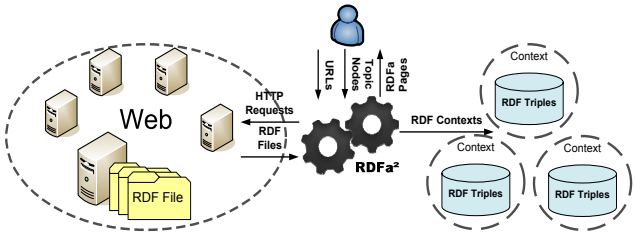


Figure 4: Context-based federated integration

One of the primary functions provided by our approach is to automatically carry out a template-based transformation of RDF to XHTML+RDFa. However, the result of the transformation will almost certainly not be in the form required by users who have diverse requirements, and consequently it is important to allows users to further edit the output. RDFa^2 allows publishers to add additional human-readable information to relevant machine-readable data. On the other hand, since each transformation will generate a template which is reusable and can be curated by users themselves, RDFa^2 also allows users to revise templates rather than revise the generated raw Web pages.

5. LINKING ANNOTATED WEB PAGES TO THE LINKED DATA CLOUD

There is one step to go before RDF triples are injected into Web pages because these embedded triples may otherwise cause provenance and trust issues. Additionally, the licence is another thing that should not be ignored especially when users attempt at reusing data from other data providers. Therefore, the enriched documents need to be associated with provenance information and linked to the Linked Data Cloud [9]. Here, we use the *Vocabulary of Interlinked Datasets* (*voiD*) [1] to describe the relationships between the annotations and the RDF contexts from which the harnessed triples are derived. Suppose the URI of the topic node is denoted by T_{uri} and the URI (or URL) of the RDF context (prove-

[7]The value of n can be any reasonable integer. Although RDFa^2 takes n to 10, by default it only just shows the top three URIs to users. It is also worth noting that blank nodes are filtered out from the set of candidates.

[8]http://prefix.cc/

[9]http://linkeddata.org/

nance) is denoted by C_{uri}. An XHTML+RDFa snippet will be automatically generated to describe the provenance of T_{uri} as follows:

```
<div about="T_uri" xmlns:void="http://rdfs.org/ns/void#" xmlns:dcterms="http://purl.org/dc/terms/">
  <span rel="dcterms:isPartOf">
    <span typeof="void:Dataset">
    <span rel="void:dataDump" resource="C_uri"/>
    </span>
  </span>
</div>
```

A increasing number of ready-to-reuse linked datasets have been published on the Web and full-fledged Linked Data application is likely to employ more than one data source. These data sources can be actually taken as different graphs (or contexts in this paper). Users are allowed to harness resources derived from different contexts based on our approach which can handle the possible conflict NSs declarations through our renaming mechanism as mentioned in Section 4.

6. EXPERIMENTS AND CASE STUDIES

RDFa^2 is the prototypical implementation for our approach. In this section, we experimented with our approach and showed the preliminary performance of RDFa^2 using Apache Tomcat installed on a PC with a Pentium®D 3.00GHz × 2 CPU and 1 GB RAM.

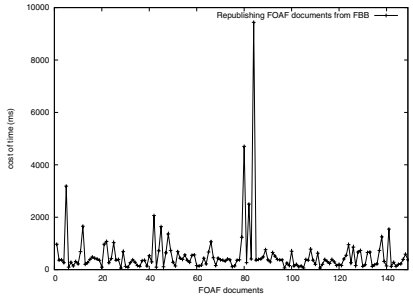
Online profiles have been widely used by various Web sites for managing user identification. FOAF is currently one of the most widely used profile vocabulary for RDF on the Web. RDFa^2 can help users inject their FOAF triples into their online profile documents such as homepages. We collected 324 FOAF documents (without considering dead links declared already on the homepage) from *FOAFBulletinBoard* (*FBB*) [10] and 146 FOAF documents from *W3C RDF Harvester Starting Point* (*WRDFHSP*) [11] respectively. These two sites are separate Wikis for bootstrapping a community in which any users are allowed to contribute FOAF documents collaboratively. Finally we got 149 and 63 valid FOAF documents in total from *FBB* and *WRDFHSP* respectively and republished them with RDFa^2 thereafter. Table 1 shows the results of retrievals of FOAF documents collected from the above two sites.

Table 1: FOAF document retrieval on *FBB* and *WRDFHSP*

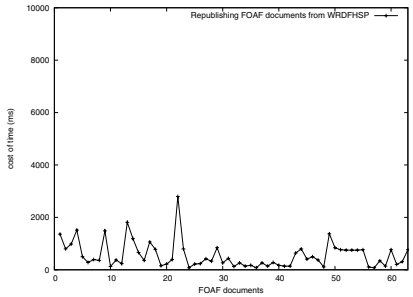
Dataset		403	404	406	503	invalid
FBB	*N*	6	72	9	1	59
	P	2.74%	22.60%	1.37%	.68%	12.33%
WRDFHSP	*N*	4	33	2	1	18
	P	1.85%	22.22%	2.78%	.31%	18.21%
Dataset		*UC*	*UKH*	*OOM*	valid	-
FBB	*N*	9	18	1	149	-
	P	6.16%	8.90%	2.05%	43.15%	-
WRDFHSP	*N*	9	13	3	63	-
	P	2.78%	5.56%	.31%	45.99%	-

In this table, by "invalid", we mean these URLs indicate FOAF documents published in an unrecommended way (e.g., FOAF documents have syntax errors or involve deprecated syntax which can not be accepted by the up-to-date RDF parser). 403, 404, 406 and 503 denote the numbers of retrievals that caused HTTP 403, 404, 406 and 503 errors respectively. *UC* denotes the numbers of retrievals that caused unconnected errors and *UKH* denotes the ones caused unknown-host errors. A few FOAF documents contain too many triples to be loaded into our parser and the number of these documents is denoted by *OOM*. *N* and *P* denote the number of retrievals and the corresponding percentage respectively. We see in this table that 54.01% of documents on *FBB* and 56.85% of documents on *WRDFHSP* do not contain valid FOAF information.

Figure 5 depicts the costs of time on republishing FOAF documents collected from the above two sites on XHTML pages with embedded RDFa. On average, 98.58% of valid documents on both sites can be transformed via RDFa^2 within 3 seconds.



(a) *FBB*



(b) *WRDFHSP*

Figure 5: Republishing FOAF documents on *FBB* and *WRDFHSP*

Figure 6 describes a screenshot within the process of republishing the FOAF document of Sir Tim Berners-Lee with RDFa^2.

RDFa^2 can be used for republishing RDF vocabularies on Web pages as well. Since there is no central repository of vocabularies on the Semantic Web [12], we retrieved RDF vocabularies in terms of NSs collected from *Ping The Semantic Web (PTSW)* [13] and *PCC* respectively. The time costs of republishing retrieved RDF vocabularies on XHTML pages with embedded RDFa are also calculated within the experiment. For 20 out of 249 vocabularies on *PTSW* as well as 5 out of 165 vocabularies on *PCC*, no results were generated after the running of the program because these vocabularies do not contain any class or property declarations. On average, however, 93.96% of successfully retrieved vocabularies can be transformed via RDFa^2 within 3 seconds. Due

[10] http://wiki.foaf-project.org/w/FOAFBulletinBoard
[11] http://esw.w3.org/AnRdfHarvesterStartingPoint
[12] http://vocamp.org/wiki/Where_to_find_vocabularies
[13] http://pingthesemanticweb.com/

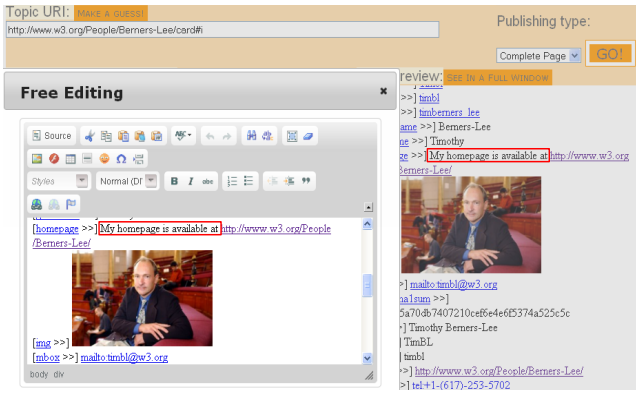


Figure 6: Screenshot for FOAF document republication with RDFa[2]

to the limited space of this paper, the statics for vocabulary retrievals and republication are not shown here.

7. RELATED WORK

FOAFr [14] allows users to convert their FOAF documents into XHTML pages with RDFa automatically. It is focused on the FOAF vocabulary [15] and can not be used for converting RDF documents described by other vocabularies. Likewise, *FOAF.Vix* [16] is a visualizer and relation explorer for FOAF documents and also dedicated to the FOAF vocabulary. It provides RDF documents in RDF/XML and Web pages containing RDFa with visualizations which have no embedded meta information. *Irs* [17] allows users to query with a specific URI for the triples containing this URI itself. These triples will be exported as XHTML+RDFa on the result page but they are rendered as a triple list and users do not have a chance to revise any content. Our approach provides users with previews of final pages on the fly so they can get real WYSIWYG experiences when doing the transformation. *Mle* [18] is able to transform mailing list archives into XHTML+RDFa pages and it is also focused on a particular vocabulary SIOC [19]. Therefore, users cannot do the annotation using other vocabularies or revise any content on the generated Web pages. *RDFohloh* [20] can provide *Ohloh* [21]'s information serialized in XHTML+RDFa, RDF/XML and N3. It is however targeted at describing projects on *Ohloh* and is subject to the terms of *Ohloh*'s APIs. *Drupal 7* allows developers to generate templates for associating RDFa with *Drupal* elements such as content types and fields [4]. However, it has not offered a solution to users for associating RDFa with more open content such as free texts created by themselves. *GoodRelations* [5] provides the *GoodRelations Annotator* [22] as well as the *Rich Snippet Generator* [23] later on, both of which assist users in creating RDFa snippets for their business or products using the particular *GoodRelations* vocabulary. By filling slots in a provided template, the user will get an RDFa snippet generated using XSLT. Our approach is not domain specific and allows users to generate RDFa snippets using any vocabularies in the RDF data model. *RDF2RDFa* [6] also allows users to copy and paste RDFa snippets generated from input RDF documents. This copy-and-paste method makes the original RDF content transparent to users so it is difficult if not impossible for users to reuse human-readable content from the original RDF documents.

[14] http://sw.joanneum.at:8080/foafr/
[15] http://xmlns.com/foaf/spec/
[16] http://foaf-visualizer.org/
[17] http://143.224.254.32/irs/
[18] http://sw.joanneum.at/mle/xplore.php
[19] http://rdfs.org/sioc/spec/
[20] http://rdfohloh.wikier.org/
[21] http://www.ohloh.net/
[22] http://www.ebusiness-unibw.org/tools/goodrelations-annotator/en/
[23] http://www.stalsoft.com/grsnippetgen/

8. CONCLUSIONS AND FUTURE WORK

Web-based knowledge sharing for community formation requires online content that can be understood by both human and machines in a more distributed manner. In this paper, a generic approach is proposed to assisting content publishers in injecting triples derived from existing RDF or RDFa documents into their Web pages. RDFa[2] has been implemented as an on-line service based on this approach and the experiments show that this tool can help publishers republish their triples in the XHTML+RDFa serialization with little manual intervention. At the time of writing, XHTML+RDFa 1.1 [24] and HTML+RDFa 1.1 [25] W3C working drafts have been released and continue to be refined, our goal is to make RDFa[2] harness new features compatible with the up-coming W3C recommendations.

9. ACKNOWLEDGMENTS

Many thanks to Prof. Ewan Klein, Dr. Giovanni Tummarello, Dr. Renaud Delbru and Thomas Schandl for their invaluable comments.

10. REFERENCES

[1] K. Alexander, R. Cyganiak, M. Hausenblas, and J. Zhao. Describing linked datasets - on the design and usage of voiD, the "vocabulary of interlinked datasets". In *WWW'09 Workshop on Linked Data on the Web*, volume LNCS 5823, pages 763–778. Springer, 2009.

[2] X. Bai, R. Delbru, and G. Tummarello. RDF snippets for Semantic Web search engines. In *ODBASE'08*, volume LNCS 5332, pages 1304–1318. Springer, 2008.

[3] X. Bai, W. Vasconcelos, and D. Robertson. OKBook: Peer-to-peer community formation. In *ESWC'10*, volume LNCS 6089, pages 106–120. Springer, 2010.

[4] S. Corlosquet, R. Delbru, T. Clark, A. Polleres, and S. Decker. Produce and consume linked data with Drupal! In *ISWC'09*, volume LNCS 5823, pages 763–778. Springer, 2009.

[5] M. Hepp. An ontology for describing products and services offers on the Web. In *EKAW'08*, volume LNCS 5268, pages 332–347. Springer, 2008.

[6] M. Hepp, R. García, and A. Radinger. RDF2RDFa: Turning RDF into snippets for copy-and-paste. In *ISWC'09 Posters and Demonstrations Track*, 2009.

[7] X. Zhang, G. Cheng, and Y. Qu. Ontology summarization based on RDF sentence graph. In *WWW'07*, pages 707–716. ACM, 2007.

[24] http://www.w3.org/TR/2010/WD-xhtml-rdfa-20101109/
[25] http://www.w3.org/TR/rdfa-in-html/