Data 2 Documents: Modular and Distributive Content Management in RDF

Niels Ockeloen(✉), Victor de Boer, Tobias Kuhn, and Guus Schreiber

The Network Institute, VU University Amsterdam, De Boelelaan 1081,
1081 HV Amsterdam, The Netherlands
{niels.ockeloen,v.de.boer,t.kuhn,guus.schreiber}@vu.nl,
http://www.networkinstitute.org, http://wm.cs.vu.nl

Abstract. Content Management Systems haven't gained much from the Linked Data uptake, and sharing content between different websites and systems is hard. On the other side, using Linked Data in web documents is not as trivial as managing regular web content using a CMS. To address these issues, we present a method for creating human readable web documents out of machine readable web data, focussing on modularity and re-use. A vocabulary is introduced to structure the knowledge involved in these tasks in a modular and distributable fashion. The vocabulary has a strong relation with semantic elements in HTML5 and allows for a declarative form of content management expressed in RDF. We explain and demonstrate the vocabulary using concrete examples with RDF data from various sources and present a user study in two sessions involving (semantic) web experts and computer science students.

Keywords: RDF · Vocabulary · Linked Data · Web documents · Content management · Semantic Web · HTML5

1 Introduction

In this paper we present a novel approach to content management which consists of providing a method for defining and rendering documents in terms of a logical composition of document fragments specified through arbitrary web resources. The approach builds on the semantic notions in HTML5 and on content in the form of Linked Data.

A still growing percentage of websites uses a Content Management System (CMS) to organise and manage their content; more than 44 % in July 2016[1] compared to 40 % two years earlier [16]. There are many CMS implementations, both open source and proprietary, the most popular currently being WordPress[2], Joomla[3] and Drupal[4]. These systems are all imperative software solutions that

[1] http://w3techs.com/technologies/overview/content_management/all, retrieved 5 July 2016.
[2] http://www.wordpress.com.
[3] http://www.joomla.org.
[4] http://www.drupal.org.

E. Blomqvist et al. (Eds.): EKAW 2016, LNAI 10024, pp. 447–462, 2016.
DOI: 10.1007/978-3-319-49004-5_29

have their own specific implementation details and database model. Therefore, sharing fragments of content between these systems is hard, and can only be accomplished by using special convertors or plugins, which either perform an offline migration[5] or depend on popular web feed formats (e.g. RSS) for live content sharing.

Though there are standards considering web documents as a whole, traditionally there have been no clear standards for dealing with fragments of documents in general. However, standards have been developed to share metadata, starting with the Meta Content Framework [9]. MCF was not widely adopted, but can be considered an ancestor of the Resource Description Framework (RDF) [13], now a major cornerstone of the Semantic Web. The number of RDF data sets that constitute the Linked Data Cloud has more than tripled between 2011 and 2014, with over a thousand data sets available containing billions of triples [20]. However, using this data in web documents is not as trivial as doing regular web content management.

In this paper we fold both these problems into one in an effort to solve them with a single solution: a novel way of doing content management expressed in RDF. By expressing all content for a web document in RDF, we provide a way of sharing heterogeneous web content between sites. By doing so, there is essentially no technical difference between regular content and other existing Linked Data. Hence, we also provide a way of including Linked Data in web documents. The main question is how to do this in a way that not only allows for the sharing and re-use of the content itself, but also the effort put into tasks such as data selection and the rendering of selected data into logical document sections.

2 Requirements and General Approach

The uniqueness of our approach is that we do not devise yet another tool to incorporate Linked Data into an existing web platform or system. Instead we turn the technology stack around; using RDF and Linked Data principles as the basis for doing content management on the web. Using this approach, we bring to the web of documents what Linked Data in general brought to the world of databases: eliminating the traditional boundaries for doing content management between web documents coming from different content owners, domains, servers and physical locations.

RDF-based Linked Data [5] has all the necessary properties to form the basis of dealing with fragments of content on the web. Any type of data can be contained, hence also fragments of web content. It has the ability to uniquely identify fragments and the ability to retrieve specific fragments using dereferencing. Furthermore, RDF is well known for its alignment abilities, making a solution expressed in RDF compatible with potential other RDF based approaches to content management. To use RDF as the basis for doing content management

[5] Example of a Joomla to WordPress migration plugin: https://wordpress.org/plugins/fg-joomla-to-wordpress/.

and sharing fragments of content on the web, we formulated the following set of requirements which the solution should meet:

- It should be built upon web standards;
- It should be able to handle heterogeneous Linked Data from multiple sources;
- It should have a separation of concerns with respect to how data is structured, selected and rendered, in the best traditions of the Web;
- It should allow for the sharing of not only content, but also the knowledge and settings involved in selecting and rendering data;
- It should be modular and distributable by design, so that bits and pieces can easily be obtained, combined and re-used on a mix-and-match basis;
- It should have a minimal set of implementation constraints;
- It should be exchangeable with other, similar approaches;
- It should be relatively easy to learn an use for a broad audience of web-contributors such as web developers and content owners;
- It should facilitate easy web page design;
- It should be doable to edit configuration and operation details by hand, in the same sense that HTML, CSS and other fundamentals can be edited by hand *if needed*, even though more elaborate GUIs exist.

To minimise the amount of infrastructure needed and rely on Linked Data standards as much as possible, we employ one of the most basic Linked Data principles to facilitate the actual managing of content: the *dereferencing* of resources. In that approach, the placement of a fragment of content, i.e. an article, is essentially done by creating an RDF triple with the IRI of the article as subject. In order to retrieve the article and render it in the document, the subject IRI is dereferenced. This principle eliminates the difference between using content within one or across sites. Obviously, there are several things we need to know in order to include the article in the document, e.g. which properties of the dereferenced resource to use, how to render the article in HTML and its positioning within the document. In order to express this information, a vocabulary can be used that is interpreted by a *general parsing script*. For this purpose, we defined the Data 2 Documents (d2d) vocabulary. We also implemented a *reference implementation* of the general parsing script to test and evaluate the vocabulary. The script interprets the d2d vocabulary elements, but is not tailored to any specific (type of) website, design or data set. The script is available on GitHub[6]. In order to meet the specified requirements, our approach has the following properties:

- All information regarding data selection, article, section and document composition and rendering are expressed in RDF which facilitates cross-site content sharing and alignment of settings with other RDF-Based systems;
- It is *declarative*, meaning that all knowledge with respect to how document composition should take place is contained in data, rather than functions;
- It provides several *abstraction layers* that are essential to perform proper content management, such as the composition of data into logical chunks of content, i.e. articles and sections, that can be reused across multiple pages and sites. The rendering of these chunks forms another abstraction layer;

[6] https://github.com/data2documents/reference-implementation-php.

- It has a *modular* setup which allows documents to be composed by choosing from various 'article definitions' that define which properties to use from a given RDF resource, which in turn can be rendered by choosing from various 'render definitions' that match the particular 'article definition';
- It is *distributable* as all content and settings are dereferenceable linked data;
- It defines a clear relation between selected data properties and semantic elements in HTML5;
- It provides a declarative template solution to facilitate web page design;
- It poses no restrictions on document structure or layout;

3 The Data 2 Documents Vocabulary

At the heart of our approach lies the d2d vocabulary, that resides in the d2d namespace http://rdfns.org/d2d/. We provide several examples of its use at http://example.d2dsite.net, including site specific content as well as linked data from external sources. A basic 'Hello World' example is given and described in detail, as well as more elaborated examples. All source RDF and template files for all examples can be viewed using an online file browser and editor at http://example.d2dsite.net/browser/. To view the contents of a file, right-click it and choose 'Edit file'. Our project website also uses d2d, and can be found at http://www.data2documents.org.

The d2d vocabulary builds upon and extends notions of HTML5[7], namely the semantic sectioning of content and the separation of a *content layer* and *style layer*. We do so by adding two additional abstract layers: that of *re-usable content*, i.e. beyond a single document, and that of *re-usable rendering* for that content. Furthermore, the d2d vocabulary is aligned with semantic document elements in HTML5 such as 'Article' and 'Section'. According to the HTML5 specification, Sections and Articles can be nested. The difference between the two being that an Article is a 'self-contained' fragment of content that is "independently distributable or reusable, e.g. in syndication".[8] How that syndication is performed in practice, is out of scope for HTML5. Due to its foundation on RDF, this distribution and re-use can be achieved using d2d and dereferencing. When we use the terms 'article' or 'section', we refer to their meaning in HTML5 (Fig. 1).

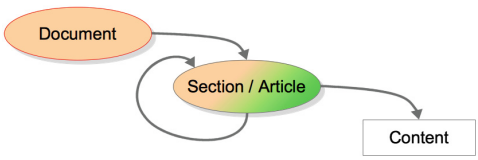

Fig. 1. A web document can be seen as a hierarchy of nested sections and articles

[7] http://www.w3.org/TR/html5-diff/#new-elements.
[8] http://www.w3.org/TR/html5/#the-article-element, retrieved 12 July 2016.

The main design choices for the d2d vocabulary are based upon the requirements discussed above and the aspects that are needed for performing web content management based on RDF. In this setup, each RDF resource can potentially be used as the source for an article or section in a web document. To facilitate this, the main tasks of the d2d vocabulary are to:

- Model the knowledge involved in selecting properties of a given RDF resource that is used as article or section;
- Model the knowledge involved in rendering the selected data properties in HTML, i.e. how to couple selected properties to HTML5 semantics;
- Model the knowledge involved in the creation of 'documents', i.e. document specific properties and the composition of articles and sections;
- Model all the above knowledge in such a way that it can be shared and re-used as small modules for specific RDF resources, articles, sections, etc.

To accomplish these tasks, the d2d vocabulary can be used to define 'Article Definitions', 'Render Definitions' and 'Documents', which are all dereferenceable RDF resources. Actual content that is to be used in the document can also be retrieved using dereferencing, as well as by using SPARQL queries (Fig. 2).

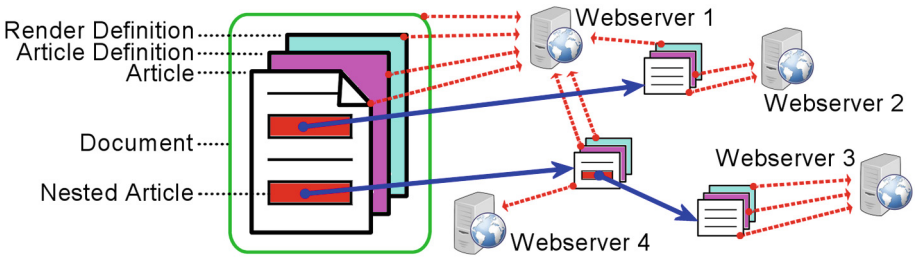

Fig. 2. Symbolic overview of the structuring of a d2d web document. The document contains one root Article, which can potentially be any RDF resource. Which properties form that resource are selected for the article is determined by the Article Definition, and how the article is rendered in HTML5 is determined by the Render Definition. The root article contains nested articles, for which separate definitions are specified. The content as well as the definitions can reside on different web servers, e.g. a content provider can provide them along with the data, or an alternative render definition can be created for a third party article type.

3.1 Main Elements of the D2D Vocabulary

Key concepts of the vocabulary are `Document` and `Section`. `Document` refers to the web document as a whole and consists of a hierarchy of nested `Sections`. One `Section` -or more precisely an `Article` which is defined as a subclass of `Section`- is the root of this hierarchy. Each `Section` contains one or more `Fields` which are small fragments of content that together make up the `Section`. How many `Fields` a `Section` or `Article` has, and of which kind, is specified by a

`Section Definition` that bundles multiple `Field Specifications`. How that `Section` or `Article` is rendered in HTML is specified by a `Render Definition`. Below is a description of the most important elements of the d2d vocabulary. A more detailed description can be found at our project website[9].

d2d:Document - Represents the web document as a whole, specifying the main render definitions to use and general document properties such as title and various meta fields. Furthermore it indicates the root article to use.

d2d:Section - Refers to a fragment of content that can be part of a web document. However, in many cases sections are not defined explicitly as external resources may be used to provide content for the section. Instead, within d2d, one can indicate that a selected resource is to be *used* as such within the `Field Specification` that selects it. `d2d:Section` has subclasses with additional semantics such as `d2d:Article`; a self-contained section.

d2d:SectionDefinition - Defines which properties related to a given resource should be used as content for a `Section`, by bundling a number of `Field Specifications`. Indicates the RDF classes that it fits to, i.e. the classes that can be used to act as the data source for a `Section`.

d2d:FieldSpecification - Specifies how data should be selected for a field that is part of a `Section`. Has a property `mustSatisfy` that either directly specifies a predicate to select data for use (shorthand notation), or points to a `TripleSpecification` that contains details on how to select the data. Has a property `hasFieldType` that specifies how to interpret the field in terms of HTML5 semantics.

d2d:TripleSpecification - Used to define a property path in order to select data for a field. The required predicate can be specified as well as details regarding the selected object such as its required type (e.g. xsd:String or foaf:Person) and role. The role determines how the selected object is used, e.g. as content for the field, as sort key, as SPARQL query, as query endpoint, or as a preferred `d2d:SectionDefinition` or `d2d:RenderDefinition` for a nested `Section`.

d2d:RenderDefinition - Defines how a `Section` should be rendered. Has a property `hasTemplate` that can either point to a file containing an HTML5 (sub) template, or a literal holding the actual template. Render definitions are optional; if not specified, fields are rendered in an HTML5 element according to their field type, while additional styling can be applied using CSS.

3.2 Interpretation of the Vocabulary

To provide insight in the vocabulary, we describe the interpretation of a document while referring to the capital letters in Fig. 3. The d2d `document` resource specifies an RDF resource that is to be used as the root `Article` (A) for the document through the `d2d:hasArticle` property. It also specifies one or more 'preferred' `Section Definitions` that define how specified RDF resources should

[9] http://www.data2documents.org/documentation#vocabulary.

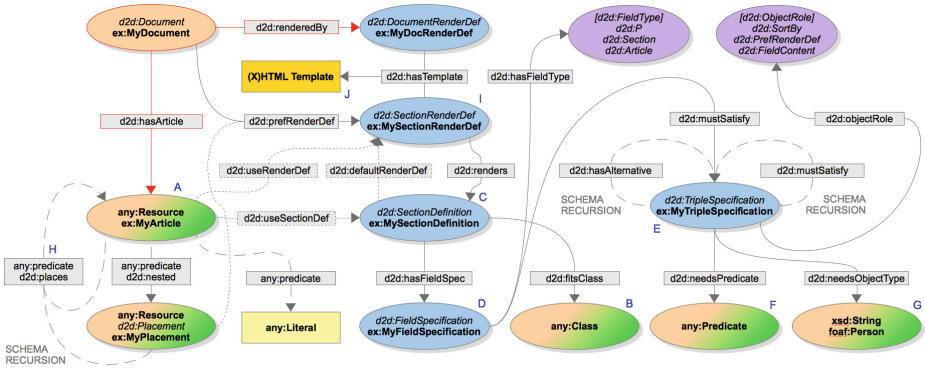

Fig. 3. Schematic overview of how the Data 2 Documents vocabulary is used to select content. Ovals represent RDF resource instances, with their class types indicated in italics. The orange ovals indicate resources that are always specifically created for a d2d web document; hence only the `Document` class itself. The orange/green ovals indicate resources that could be either specifically created for the web document, or be resources in an existing Linked Data set not specifically created for the document. The blue ovals are resources that are instances of d2d classes such as the `Section` and `Render Definitions`. Lastly, the purple ovals are classes from the vocabulary itself that are used to indicate the desired role or field type. A high resolution version of this figure is also available at http://www.data2documents.org/ppre/d2d-schema-0916.png. (Color figure online)

be used as `Article` or `Section`, i.e. which properties should be selected. Exactly which `Section Definition` matches the class of the particular RDF resource is specified through the `d2d:fitsClass` property (B). The matching `Section Definition` (C) contains one ore more `Field Specifications` (D) that specify a particular field for the article. How data for that field should be selected is indicated by the `Triple Specification` (E) that specifies the predicate (F) that the resource *acting as article* should have in order to satisfy the particular field. Optionally, the required type of object for that predicate can be specified (G), and by chaining several `Triple Specifications` a longer property path (H) and/or alternatives can be specified. The data that is to be selected can be a literal, e.g. for a paragraph of text, or yet another resource that is to be processed as a *nested* `Section` or `Article`. Finally, a matching `Render Definition` (I) defines how to render the selected data, optionally using a (sub)template (J) specified by the `Render Definition`. These steps are repeated recursively for each nested `Section` or `Article`, until the complete documents has emerged.

3.3 Declarative Template Solution

In order to gain more control over document rendering, d2d includes a declarative template solution. Specifying templates is optional, as content is already associated with HTML5 semantic elements in the article definition and can be rendered

on that basis, while further styling can be done using CSS. (Sub)templates can be nested in a single HTML file that loads and renders individually, to facilitate design. Listing 1.1 provides a basic example. Due to space limitations we cannot provide more extensive examples in this paper. However, more examples and descriptions can be found on our project website, section documentation[10].

```
<d2d:Template d2d:for={article definition IRI}>
  <d2d:Field>
    <!-- Conditional HTML; placed only if field is placed -->
      <d2d:Content>
        <!-- Sample content; gets replaced; Can contain nested Templates -->
      </d2d:Content>
    <!-- Conditional HTML; placed only if field is placed -->
  </d2d:Field>
</d2d:Template>
```

Listing 1.1. Example of d2d HTML template for an article with one field. If no conditional HTML or sample content is needed, just `<d2d:Field />` will suffice.

4 Related Work

Several aspects of our work relate to other scientific work that has been carried out in the past. These aspects include tasks such as the rendering and visualisation of Linked Data, the editing of semantic content, the selection of suitable data using a set of constraints and the use of Linked Data within content management systems.

Vocabularies. Though various vocabularies exist that could be relevant for web content management such as VoID [1] for data set description, PROV [8,15] for describing the provenance of document sections and SHACL[11] for describing and validating RDF Graphs, to the best of our knowledge, no vocabulary exists to the describe the knowledge required to perform actual web content management.

Web Feed Formats. RSS 1[12], 2[13] and Atom[14] are prominent ways of sharing and re-using content across web sites; a process called syndication. Though they can be extended with terms from other name spaces, in practise this only works well for specific domains, e.g. Podcasts[15]. This is due to the fact that many general purpose parsers don't know what to do with such added fields an simply ignore them. In order to process any kind of field without the need of adding explicit support within implementations, a *meta model* is needed to describe how to interpret such fields in a specific context, as is the case in our approach.

Semantic Portals. Semantic MediaWiki [11] allows for semantic annotations to be made within unstructured web content, whereas OntoWiki [2] is form-based and organised as an 'information map' in which each semantic element is

[10] http://www.data2documents.org/documentation#templates.
[11] Shapes Constraint Language: https://www.w3.org/TR/shacl/.
[12] http://web.resource.org/rss/1.0/.
[13] http://www.rssboard.org/rss-specification.
[14] http://tools.ietf.org/html/rfc4287.
[15] RSS extended with 'itunes' elements: http://www.podcast411.com/page2.html.

represented as an editable node. They are tools for authoring semantic content, but are not build specifically to be used as general content management system and do not facilitate the live sharing of semantic content across sites.

Linked Data Rendering and Visualisation. Fresnel [18] is an OWL-based vocabulary that can be used to define *lenses* to select data from a given data graph and *formats* that define how that data should be displayed. Exhibit [12] is an AJAX based publishing framework for structured data that uses an internal data representation format and allows the data to be used in rich web interfaces using templates and pre-defined UI types such as faceted browsing. Callimachus [4] is a template based system that facilitates the management of Linked Data collections and the use of that data in web applications. Uduvudu [14] is a visualisation engine for Linked Data built in JavaScript that lets users describe recurring subgraphs in their data and indicate how these subgraphs should be visualised. Balloon Synopsis [19] is also an approach to include Linked Data in web publications running at the client side, implemented as a jQuery[16] plugin. When it comes to visualising Linked Data in general, Dadzie and Rowe [7] provide a survey of approaches. What makes our approach different is the separation of tasks such as the selection of data elements to form logical documents sections and the rendering of these sections into documents, set up in a way that is completely declarative, modular and distributable in order to facilitate the sharing and re-use of not only the actual content, but also the definitions of what data to select and how to render it in the document. RSLT [17] is a transformation language for RDF data that uses templates associated to resource properties.

Content Management Systems. Drupal has an RDF library to expose information as Linked Data through RDFa [6]. OntoWiki CMS [10] is an extension to OntoWiki that combines several other tools such as the OntoWiki Site Extension and Exhibit to facilitate the use of OntoWiki data in web documents and a dynamic syndication system. The Less template system [3] allows Linked Data that is accessed through dereferencing or SPARQL queries to be used in text-based output formats such as HTML or RSS. It can be used in collaboration with existing content management systems such as WordPress and Typo3[17], however it uses data property names directly in its templates thus provides no separation between data selection and data representation. These plugins and tools focus on the production and consumption of RDF Data within existing systems, but they do not exploit the intrinsic Linked Data properties to facilitate web content management, nor do they store information with respect to the actual content management such as document composition in RDF; it is maintained in implementation specific database systems, which limits interoperability.

5 Evaluation

To evaluate our approach, we performed two experiments: With the first one, we wanted to test the usability and usefulness for expert users, resembling potential

[16] http://jquery.com.
[17] http://typo3.org.

power-users of our system. For this we recruited users who are proficient in Linked Data and Web development techniques. With the second experiment, we wanted to test our approach on users who are technically minded but have limited knowledge and practical experience with Linked Data and Web development. These participants should resemble potential casual users of our system as well as beginners who are trying out these new technologies, but are not yet very familiar with them. For this second experiment, we recruited a larger group of students in Computer Science-related Master programs.

5.1 Design of the Experiments

Both experiments have the same basic structure: The participants first received a brief introduction into the basic concepts of the Data 2 Documents technique. This was done via a general presentation and demo of about 10 min. Then the participants received the detailed instructions, where they were first asked to register and login into a system were each participant was given a separate subdomain and web server instance with an elementary online source code editor. Figure 5 shows a screenshot of that source code editor. In this editor, the participants found two directories: a read-only directory with a working example[18], and a writable directory that was initially empty and where the participants should create their own website based on the example website[19]. After completing the given tasks, the participants had to fill in a questionnaire asking them about their background, skills (Fig. 4), experiences during the experiment, and their opinions on d2d. Participants were given 9 tasks:[20]

- **Task 1:** Creating a new d2d document by copying from an example
- **Task 2:** Adding a missing introduction article to the new document
- **Task 3:** Changing the title of the introduction article
- **Task 4:** Linking and thereby including an existing FOAF profile as article
- **Task 5:** Creating a new *comment* article and including it in the document
- **Task 6:** Linking and thus including comment articles of other participants

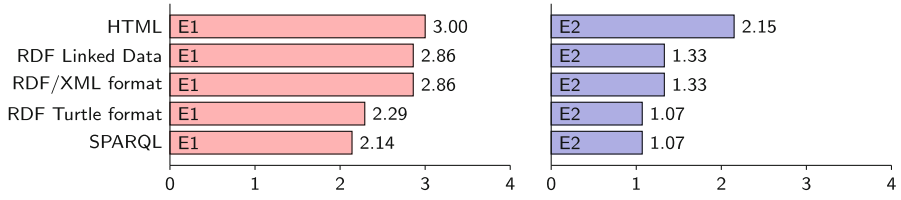

Fig. 4. The average skill level of the participants for experiments E1 and E2

[18] Example site: http://kmvuxx.biographynet.nl.
[19] Participant site after completing the evaluation: http://kmvu03.biographynet.nl.
[20] The detailed instructions can be found here: http://biographynet.nl/assignment/.

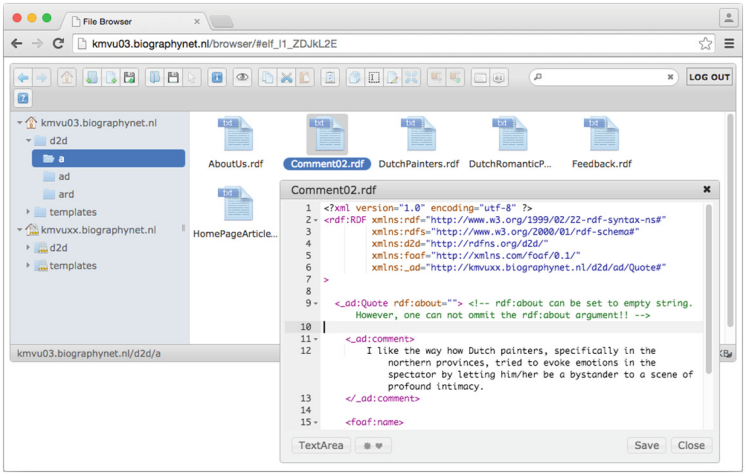

Fig. 5. The online browser and text editor that was used for the experiments. Though all tasks consisted only of elementary text editing, we had to provide a way of browsing and editing the text files, which reside on a web server.

- **Task 7:** Changing an existing listing to show Dutch Painters instead of Prime Ministers of The Netherlands
- **Task 8:** Extending the article definition, render definition and template to show an additional field (creation date) in a listing of paintings
- **Task 9 (optional):** Create a new listing of people using a DBpedia category

For the first experiment, we recruited among employees of VU University Amsterdam as well as their friends who work on general web development and/or Linked Data. The participants of the second experiment were Master students from the *Knowledge and Media* course that was given at the same university during fall 2015.

5.2 Results of the Experiments

For the first experiment, we managed to recruit a relatively small group of 7 participants, but for the second experiment we got a large group of 73 students. Two of the seven participants of the first experiment were female (29 %), which is almost the same rate as for the second experiment, for which 30 % were female. The average age was 35.3 years for the first experiment and 24.5 years for the second. The students of the second experiment were mostly enrolled in the Master programs *Information Science* (76.7 %) and *Artificial Intelligence* (16.4 %).

The left hand side of Fig. 6 shows the rates at which the different tasks were successfully completed by the participants of the two experiments, as revealed by inspecting the resulting files on the server instances after the experiment. These rates are very high at close to or over 85 % for all the tasks up to and including Task 5 for both experiments, and they did not go below 50 % except for the bonus Task 9 (40 %).

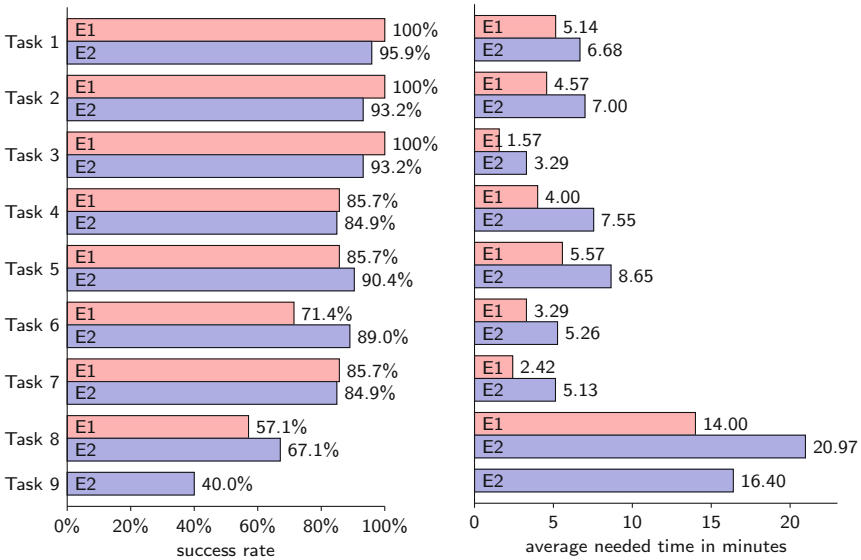

Fig. 6. Success rates (left) and average required time (right) for the different tasks of the two experiments.

For tasks 6 to 8 the success rates are — surprisingly — lower for the experts than for the students, considerably so for tasks 6 and 8. Both groups performed very well, but for some reason the experts were not as good in completing these tasks as the students, even though the latter have a lower skill level.

The solution to this puzzle is presented in the chart on the right hand side of Fig. 6, which plots the average amount of time (in minutes) the participants needed to complete the respective tasks. Over all tasks, the experts spent on average considerably less time than the students. Together with the data for the task success rates presented above, this seems to suggest that the experts did not try as hard as the students. Compared to the students, the experts seem to have been more interested in finishing the experiment in a relatively quick fashion, whereas the students seem to have been more committed to the tasks and were willing to spend more time to complete them.

In absolute terms, the numbers of both settings look very promising. Even the students needed on average only a bit more than one hour to complete the tasks 1 to 8. Taking into account that building or adjusting a website with dynamic and complex content is inherently a challenging task and that users will become more efficient over time when using the same tool, these results seem to indicate that our approach is indeed useful and appropriate.

Table 1 shows an aggregation of the responses of the participants to seven statements (S1–S7) in the questionnaire. They were asked whether or not they agree with these statements on a scale from strongly agree (1) to strongly disagree (5). These statements ask about the participants' opinions with respect

Table 1. Answers from the participants of the two experiments about the degree to which they agree with the given statements about d2d. (* = significant)

Experiment:	E1	E2							
Statement	Avg.	←agree disagree→					Avg.	strong/weak agree (≤2)	not disagree (≤3)
		1	2	3	4	5		p-value	p-value
S1: "d2d seems to be a suitable approach to perform general Web Content Management such as the creation, sharing and placing of content articles"	2.14	17	40	9	6	1	2.10	* $<10^{-6}$	* $<10^{-12}$
S2: "d2d seems to be a suitable approach to eliminate the traditional boundaries for Content Management between separate web sites, documents, and domains"	1.57	24	29	13	5	2	2.07	* $<10^{-4}$	* $<10^{-12}$
S3: "d2d makes it easy to share content between separate web sites/documents/domains"	1.43	28	22	16	5	2	2.05	* 0.0011	* $<10^{-12}$
S4: "d2d seems to be a suitable approach to use Linked Data in web documents"	1.29	29	29	7	8	0	1.92	* $<10^{-6}$	* $<10^{-11}$
S5: "Manually editing d2d definitions is not significantly harder to do than manually editing HTML"	2.29	25	15	18	13	2	2.34	0.2414	* $<10^{-6}$
S6: "I would consider using d2d, if I have to develop a general website in the future"	3.29	11	22	24	11	5	2.68	0.8254	* $<10^{-6}$
S7: "I would consider using d2d, if I have to develop a website in the future that makes use of Linked Data"	1.71	25	28	9	9	2	2.11	* $<10^{-4}$	* $<10^{-9}$

to the suitability of the approach for different goals (S1–S4), how it compares to plain HTML documents (S5), and whether they would consider using the framework for themselves in the future (S6 and S7). Due to the smaller number of participants, only the average values are shown for experiment 1, but more details are given for the second experiment. All average values are on the *agree* side (which for all statements means in favor of d2d) with the exception of the response from the experts (experiment E1) on statement S6 (but they do strongly agree with the more specific statement S7).

For the second experiment with a larger number of participants, we can make some more detailed analyses. If we lump together *strong agree* and *weak agree* (i.e. ≤ 2), we get a majority of the responses in this area for all statements except S6. To test whether these majorities are not just a product of chance, we can run a statistical test. Our null hypothesis is that when users are asked to select between *strong/weak agree* (≤ 2) on the one hand, and any other option (i.e. *neutral* or *strong/weak disagree*: >2) on the other, they would tend towards the latter or at most have a 50 % chance of selecting *strong/weak agree*. We use a simple one-tailed exact binomial test to evaluate this hypothesis for each of the statements with the data from experiment 2. The results can be seen in Table 1 and they show that we can reject this null hypothesis for all statements except S5 and S6. We have therefore strong statistical reasons to assume that the tendency towards *strong/weak agree* for statements S1, S2, S3, S4, and S7 is not just the product of chance. In a next step, we can soften our previous null hypothesis a bit by adding *neutral* to the lumped-together area, and see whether users have a tendency towards saying *strong/weak agree or neutral* (i.e. ≤ 3). As shown in Table 1, this softer null hypothesis can be rejected for all statements. All data collected and additional charts are available on our project website[21].

6 Conclusions and Future Work

In this paper we described Data 2 Documents (d2d): A vocabulary for doing content management in a declarative fashion, expressed in RDF. The vocabulary is accompanied by a reference implementation that interprets it to create rich web documents. Our results show that participants do not disagree that manually editing d2d definitions is not significantly harder to do than manually editing HTML. We think that this is an impressive result if we consider how much more powerful d2d is, compared to plain HTML. Moreover, as S7 shows, a majority of users would consider using d2d in the future to develop websites making use of Linked Data.

Here, we described Data 2 Documents in its fundamental form, having all content data and definitions as editable XML/RDF files. As future work we plan to develop a sophisticated GUI for working with d2d. But we can do so with the assurance that users are able to fall back on elementary editing skill should it be required. We also plan to port our reference implementation to SWI Prolog[22] for large scale applications to run directly on the ClioPatria[23] semantic web server. A port to Javascript is also planned to browse d2d web documents directly on the client side by requesting the raw RDF data and in order to include d2d defined articles in non-d2d web documents.

We see Data 2 Documents as a first step to bring together the largely separated networks of documents (the Web) and data (Linked Data), for Web users to benefit from the increasing amount of structured data. As such, we think that

[21] http://www.data2documents.org/#evaluation.
[22] http://www.swi-prolog.org.
[23] http://cliopatria.swi-prolog.org.

our approach might form an important step to finally make the vision of the Semantic Web a reality.

Acknowledgments. This work was supported by the BiographyNet(http://www.biographynet.nl) project (Nr. 660.011.308), funded by the Netherlands eScience Center(http://esciencecenter.nl). Partners in this project are the Netherlands eScience Center, the Huygens/ING Institute of the Royal Dutch Academy of Sciences and VU University Amsterdam.

References

1. Alexander, K., Hausenblas, M.: Describing linked datasets-on the design and usage of void, the vocabulary of interlinked datasets. In: Linked Data on the Web Workshop (LDOW 2009). Citeseer (2009)
2. Auer, S., Dietzold, S., Riechert, T.: OntoWiki – a tool for social, semantic collaboration. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 736–749. Springer, Heidelberg (2006)
3. Auer, S., Dietzold, S., Riechert, T.: OntoWiki – a tool for social, semantic collaboration. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 736–749. Springer, Heidelberg (2006). doi:10.1007/11926078_53
4. Battle, S., Wood, D., Leigh, J., Ruth, L.: The callimachus project: RDFa as a web template language. In: COLD (2012)
5. Bizer, C., Heath, T., Berners-Lee, T.: Linked data - the story so far. Int. J. Semant. Web Inf. Syst. **5**(3), 1–22 (2009)
6. Corlosquet, S., Delbru, R., Clark, T., Polleres, A., Decker, S.: Produce and consume linked data with Drupal!. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (eds.) ISWC 2009. LNCS, vol. 5823, pp. 763–778. Springer, Heidelberg (2009)
7. Dadzie, A.S., Rowe, M.: Approaches to visualising linked data: a survey. Semant. Web **2**(2), 89–124 (2011)
8. Groth, P., Gil, Y., Cheney, J., Miles, S.: Requirements for provenance on the web. Int. J. Digit. Curation **7**(1), 39–56 (2012)
9. Guha, R.: Meta content framework. Research report Apple Computer, Englewoods, NJ (1997). http://www.guha.com/mcf/wp.html
10. Heino, N., Tramp, S., Auer, S.: Managing web content using linked data principles-combining semantic structure with dynamic content syndication. In: 2011 IEEE 35th Annual Computer Software and Applications Conference (COMPSAC), pp. 245–250. IEEE (2011)
11. Herzig, D.M., Ell, B.: Semantic MediaWiki in operation: experiences with building a semantic portal. In: Patel-Schneider, P.F., Pan, Y., Hitzler, P., Mika, P., Zhang, L., Pan, J.Z., Horrocks, I., Glimm, B. (eds.) ISWC 2010, Part II. LNCS, vol. 6497, pp. 114–128. Springer, Heidelberg (2010)
12. Huynh, D.F., Karger, D.R., Miller, R.C.: Exhibit: lightweight structured data publishing. In: Proceedings of the 16th International Conference on World Wide Web, pp. 737–746. ACM (2007)
13. Lassila, O., Swick, R.R.: Resource description framework (RDF) model and syntax specification. Technical report (1999). http://www.w3.org/TR/REC-rdf-syntax/

14. Luggen, M., Gschwend, A., Anrig, B., Cudré-Mauroux, P.: Uduvudu: a graph-aware and adaptive UI engine for linked data. In: Proceeding of the LDOW (2015)
15. Moreau, L., Missier, P., Belhajjame, K., B'Far, R., Cheney, J., Coppens, S., Cresswell, S., Gil, Y., Groth, P., Klyne, G., Lebo, T., McCusker, J., Miles, S., Myers, J., Sahoo, S., Tilmes, C.: PROV-DM: the PROV data model. Technical report, W3C (2012). http://www.w3.org/TR/prov-dm/
16. Ockeloen, N.: RDF based management, syndication and aggregation of web content. In: Lambrix, P., Hyvönen, E., Blomqvist, E., Presutti, V., Qi, G., Sattler, U., Ding, Y., Ghidini, C. (eds.) EKAW 2014. LNCS (LNAI), vol. 8982, pp. 218–224. Springer, Heidelberg (2015). doi:10.1007/978-3-319-17966-7_31
17. Peroni, S., Vitali, F.: Templating the semantic web via RSLT. In: Gandon, F., Guéret, C., Villata, S., Breslin, J., Faron-Zucker, C., Zimmermann, A. (eds.) ESWC 2015. LNCS, vol. 9341, pp. 183–189. Springer, Heidelberg (2015). doi:10.1007/978-3-319-25639-9_35
18. Pietriga, E., Bizer, C., Karger, D.R., Lee, R.: Fresnel: a browser-independent presentation vocabulary for RDF. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 158–171. Springer, Heidelberg (2006)
19. Schlegel, K., Weißgerber, T., Stegmaier, F., Granitzer, M., Kosch, H.: Balloon synopsis: a jQuery plugin to easily integrate the semantic web in a website. In: CEUR Workshop Proceedings, vol. 1268 (2014)
20. Schmachtenberg, M., Bizer, C., Paulheim, H.: Adoption of the linked data best practices in different topical domains. In: Mika, P., Tudorache, T., Bernstein, A., Welty, C., Knoblock, C., Vrandečić, D., Groth, P., Noy, N., Janowicz, K., Goble, C. (eds.) ISWC 2014, Part I. LNCS, vol. 8796, pp. 245–260. Springer, Heidelberg (2014)