Design Issues for the Development of Linked Data Content Management Systems with Reasoning Capabilities

Abstract— **The new developments in data interfaces, query languages, development tools and an increasing vision of the need for a new ecosystem of globally interconnected data, promotes the development of new frameworks for developing web sites able to publish their own structured data and connect with other repositories under the rules and precepts of initiatives Linked Open data (LOD). The paper presents a proposed framework that allows publishing and data description using local standards and vocabularies as SKOS, Dublin Core, RDFS, automatic creation of classes, properties and attributes associated with the data, while coexisting with the traditional website development based on unstructured content. Finally a framework to integrate reasoning technologies with the generation and consumption of Linked Data is described. The author also briefly describes some of the technologies used in the proposal.**

Keywords— Linked Open Data; Software Development Frameworks; Content Management Systems; OWL Reasoning Engines; OWL; RDF

I. INTRODUCCION

A. Some terms and names of technologies

Linked Open Data: Linked Data is about using the Web to connect related data that wasn't previously linked, or using the Web to lower the barriers to linking data currently linked using other methods. More specifically, Wikipedia defines Linked Data as "a term used to describe a recommended best practice for exposing, sharing, and connecting pieces of data, information, and knowledge on the Semantic Web using URIs and RDF."

RDF: The Resource Description Framework (RDF) is a family of World Wide Web Consortium (W3C) specifications originally designed as a metadata data model. It has come to be used as a general method for conceptual description or modeling of information that is implemented in web resources, using a variety of syntax notations and data serialization formats. It is also used in knowledge management applications. [1]

Angular.js: is an open-source web application framework, maintained by Google and community, that assists with creating single-page applications, one-page web applications that only require HTML, CSS, and JavaScript on the client side. Its goal is to augment web applications with model–view–controller (MVC) capability, in an effort to make both development and testing easier. [2]

Markdown: is a text-to-HTML conversion tool for web writers. Markdown allows you to write using an easy-to-read, easy-to-write plain text format, then convert it to structurally valid XHTML (or HTML). [3]

Twig: is a modern template engine developed for PHP by SensioLabs. [4]

LOD Development Framework: Components, modules, classes and objects to develop a web site or application LOD, created in any programming language to use on one or multiple operating systems.

LOD Tools: Applications, web sites or specific software modules used to manage, generate, transform or consume data with formats used by the LOD initiatives (such as RDF, XML, etc.).

B. Linked Open Data

The Web was originally designed with a hypertext vision: documents referring others via embedded links in the documents themselves [5]. As the evolution of the Internet demanded more complexity and interactivity for content, concepts and paradigms and describing the information published on the Web were improved and expanded, although maintaining the concept of "document - link - document".

The concept of structured data embedded in the information published on the Web can be found in those same beginnings [6] and interpreted in different ways in the following years (in [7] and [8]) to complete the concept of Semantic Web we have today. The concept of Linked Open Data [9] is actually one of the first practical movements in this direction: connect data instead of documents to consume them and to find other related data, and around this simple concept to create an ecosystem of applications, repositories, rules and methods to exploit the new structures and relationships in order to build knowledge.

C. Current Status of LOD on the web

Currently the map shows large LOD technology domains on the major developments focus. A summary of the distribution of these domains are shown in the following table and graph [10]:

Domain	Volumen (%)
Government	42.09 %
Geographic	19.43 %
Cross-domain	13.23 %
Life sciences	9.60 %
Publications	9.33 %
Media	5.82 %
User-generated content	0.42 %

Fig. 1: Distribution of LOD content domains, ordered by amount of triples generated

However, although most of the contents is published under the government or geographic domains, the connectivity distribution of such content with other sources is quite different [10]:

Domain	Volumen (%)
Government	38.06 %
Geographic	27.76 %
Cross-domain	12.54 %
Life sciences	10.01 %
Publications	7.11 %
Media	3.84 %
User-generated content	0.68 %

Fig. 2: Distribution of LOD content domains, ordered by number of links to other data sources

D. LOD Tools categorized by type of application

Tom Heat and Bizer in [11] define two major categories for LOD applications: *Generic Applications* and *Specific-Domain Applications*, but now (2014) is possible to differentiate between more categories, in the case of *Generic Applications*. The following list is not intended to cover all possible categories, but to provide a closer map to the current state of development of the LOD tools.

1. **Frameworks and development libraries**: Tools for developers, such as in the case of Skydata (https://github.com/egiralt/skydata) or SemanticWebBuilder (http://www.semanticwebbuilder.org.mx/) with which it is possible to code, completely or partially, a LOD website.
2. **Data & content management systems**: In this section we mention several commercial products, such as Mondeca (http://www.mondeca.com/) or others opensource/free as Drupal (https://www.drupal.org/) with its ability to generating and consuming RDF. A possible subcategory to this are the systems used exclusively for publishing and data management like CKAN (http://ckan.org/) o D2R Server (http://d2rq.org/d2r-server).
3. **LOD Wrappers** : These tools act as mediators between the "traditional" queries to websites like Crunchbase (http://www.crunchbase.com/) or institutions such as Eurostat (http://epp.eurostat.ec.europa.eu/) to post data expressed in RDF or other formats of the associated LOD. We can mention others more specialized examples such as SIOC project which has created wrappers for different Content Management Systems (CMS) such as Drupal or Wordpress to allow data exporting in RDF format. Also Triplify (http://triplify.org/) does something similar with many other CMS.
4. **Query interfaces**: This category could be describe sites devoted to allow consultations to their datasets, generally through public accessible endpoints such as implemented by Linked Data Fragments (http://linkeddatafragments.org/) or sites that offer access to their repositories with customized methods like LODRefine (http://code.zemanta.com/sparkica)
5. **LOD Browsers**: Although there is no clear trend that major browsers (i.e. Chrome, IE, Firefox) implicitly will incorporate capabilities of data recovery and navigation in short term, there are several developments in order to extend them with plugins to allow these extensions, as in the case of Tabulator (http://www.w3.org/2005/ajar/tab). On the other hand we have some online browsers fully devoted to RDF and Linked Data as LODLive (http://en.lodlive.it/) or OpenRDFBrowser (http://demo.openlinksw.com/DAV/JS/rdfbrowser/index.html)
6. **Converters and format translators**: In this category is more common to find small applications used locally to generate RDF static files that can be later published, as ConverterToRDF (http://www.w3.org/wiki/ConverterToRdf) or RDFizers (http: // simile. mit.edu/wiki/RDFizers). On the other hand DataLift (http://datalift.org/) propose a comprehensive platform for converting different sources (CSV, XML, RDFa ...) to interconnected data.

See also [12] for specific tools used in the management of multimedia content.

II. Designing Linked Data Content Managemen Systems (LDCMS)

This term has been mentioned several times (eg in [13] and [14]) to describe complete systems able to manage, publish and consume structured data, but we can´t find a complete and standard guide to consider which kind of capabilities or functional elements defines such systems, as opposed to the "traditional" Content Management Systems (CMS)

Considering that the data evolves to the side of unstructured content, we have to find models to incorporate the Linked Data to the same spaces which others have the supremacy: HTML and textual & unstructured formats. Maybe the most obvious solutions is to design new data management systems to deliver datasets from specialized and authored sources, but… what happen with the "normal" websites? What happen, for example, with a restaurant or fitness company trying to publish their recipes or daily routines for their clients? Even this kind of data could serve particular ontologies and reuse existent ones (ie. GoodRelations Ontoly (http://www.heppnetz.de/projects/goodrelations/) or BBC's Food Ontology (http://www.bbc.co.uk/ontologies/fo)

The following proposals aim to support a definition of the capabilities that define the architecture of an LOD application, from the point of view of the developer or software architect.

A. Design model for Linked Open Data Websites

This model proposes a general list of components working together to deliver content from different source: from static files to database, connecting with external repositories or ontologies to describe and extend the internal datasets.

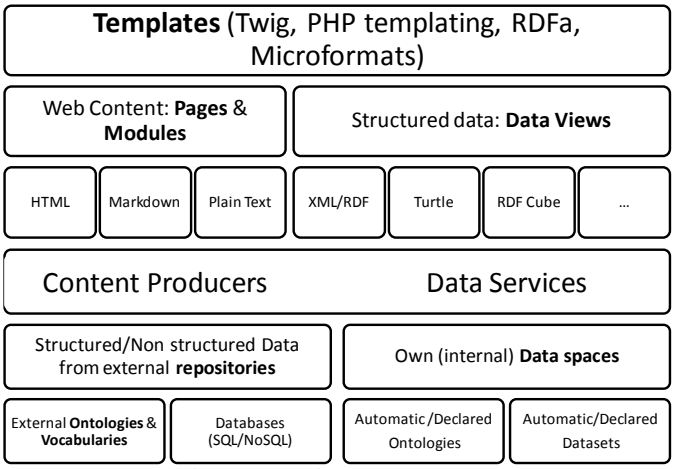


Fig. 3: General model for a LDCMS framewok

We can make an inventory of the main components in this model:

Design Component	Description
Resource	A class specifies that provides specific properties for publishing content and structured data to subclasses
Page	A resource type that perform the publication of a full content. Use the Services and Content Providers as sources for data & content. A Page is a kind of ContentConsumer and can contain Modules to build more complex outputs
Module	Generates a piece of content to place on a Page. Use the same sources of content as Page. It is also a kind of ContentConsumer
ContentProducer	Classes responsible of delivering content to ContentConsumers. The original content can come from databases, files, online sources, etc.
ContentConsumer	Classes responsible for receiving content from ContentProducers to apply some transformations or deliver to visualization on browsers.
Service	Classes that use various content sources to "inject" data or information in ContentConsumers. These classes are the bricks for some kind of interactive used to display and encapsulate information according to Web Components Specifications [15]
Template	Intermediate classes that support design-content separation, in other words: they contain the patterns needed to display the content according to media device or final format, regardless of the value or content structure.
Dataview	A class that manages the delivery or display of structured data, as opposed to Pages containing unstructured content. It can be used to defer the delivery of data in other types of browsers or clients, eg RDF files to deliver Open Linked Data browsers
VocabulariesSources	List of terms and definitions that will linked to the descriptions of the data and metadata with semantic value. The list of vocabularies generated in the same application will be part of the global list with external vocabularies.
OntologiesSources	List of ontologies and semantic schemas used to define entities, attributes and relationships used in the application dataspace
Dataset	A resource type that defines a specific instance of a data entity declared in the application
DataspaceManager	Classes responsible for managing and organizing the set of all declared datasets in the application
Controllers	These classes are responsible to manage the business logic for the application. They act as intermediate for Pages, Services and Producers (as in the MVC [16] pattern or other similar paradigms)

B. Administrative modules to support Linked Data

An inseparable part of a LDCMS are the administrative modules responsible to create, store and manage all the infrastructure under the data & content publication in a LOD application, but now it is necessary to consider the new functional requirements for LOD.

We can distinguish four new major modules for the LDCMS paradigm:

1. **Management of data definitions (DataSets)**: The system will provide capabilities that allow creation of new dataset and the assignation of its metadata, vocabularies and associated domains. These definitions could specify external data (ie. published in other websites) or internally created. It is important that the authors can select and apply vocabularies and ontologies (stored in ***VocabulariesSources*** and ***OntologiesSources*** components) associated with the area of the site. ie. *if the site publishes information on sports programs the authors should be get a full list of convenient ontologies or vocabularies about sports and TV, including the BBC's Sport Ontology* [17]. Definitions of data can also be imported or automatically extracted from the data sources. ie: *If used as a source a CSV data file, the system will propose the structure of fields and their types inferred from the format in which the values are stored*.
2. **Management of data sources**: Since the data can come from many types and volumes of different ranks - from a few data to complex databases – the system has to delegate the editing and preparation of the data to more specialized systems (MS Excel, relational databases servers, CKAN, etc.), but it is necessary that exist way to manage those connections and deliver the access as a list of sources that can then be connected to the corresponding data definitions.
3. **Publication management**: The authors should specify the type of output or processor used to output data, besides templates to be used for the distribution

of the final content. Furthermore, they should be able to manage:

a. Dates of publication and maturity
b. Version control
c. Endpoints for consultations
d. Access control for users and third parties, including web domains, IP addresses, etc
e. Delivery control based on languages, devices or browsers, including accepted formats for content (*ex. Only XML/RDF*)

4. **Management of sources of vocabularies, schemas and ontologies**: Have a list of existing and defined data types or content resources is crucial, since reuse and reference are key elements in the development of a interconnected data balanced ecosystem. The system will provide the options to indicate the domain of application and according to that definition, will find, retrieve and use external vocabularies and ontologies that enable to authors to apply the definitions and appropriate equivalences in each case.

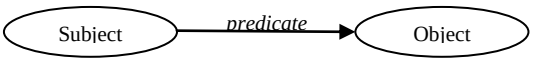
III. Integrating Reasoning Engines with LDCMS

There are many recognized commercial and open and free source software that implements inference / reasoning capabilities using OWL and RDF as main languages. These applications and can be easily integrated with LOD. Among these can be counted Jena (https://jena.apache.org/) , Virtuoso (http://virtuoso.openlinksw.com/) or Pellet (http://clarkparsia.com/pellet/) .

Considering that a RDF (triple) clause consists of:

```
( <subject>, <predicate>, <object> )
```

Or, graphically represented:



it is possible to translate this construction to clauses of first-order logic (FOL), applying the results to build inferences, to satisfy searches, to create data aggregations or in the generation of new clauses, therefore generating new knowledge. That could be a goal for a new generation of Linked Data websites and application in order to build an intelligent WWW based on evolutive data & content: the Evolutive WWW

A. *Design model extending the LDCMS framework to include reasoning capabilities*

The scheme shown in Figure 3 can be extended to the following one to incorporate reasoning capabilities provided by the engines discussed before. It should be noted that this model does not include administrative or support modules, need for a more comprehensive model.

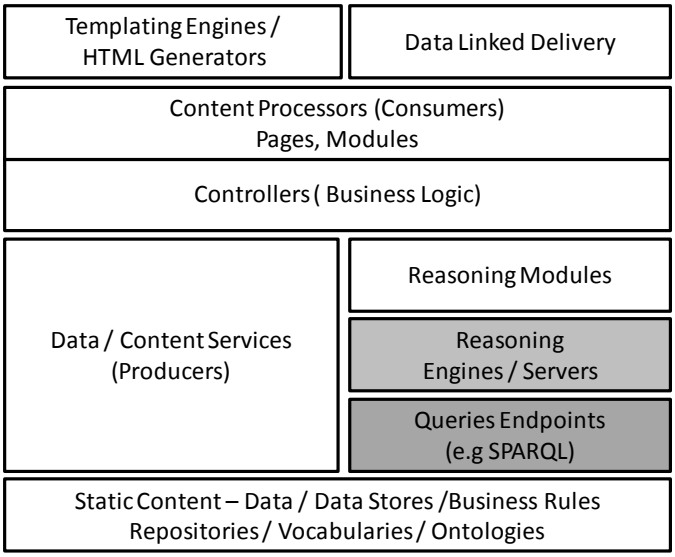


Fig. 4: General scheme for a LDCMS framework integration reasoning capabilities.

It is important also to note that the reasoning modules shown are part of the framework itself, but not the reasoning engines or the software managing the queries endpoint that can be delegated to libraries o external applications (optionally).

IV. References

[1] Wikimedia Foundation, Inc., "Resource Description Framework," Aug 2014. [Online]. Available: http://en.wikipedia.org/wiki/Resource_Description_Framework .

[2] Wikimedia Foundation, Inc., "AngularJS," Sep 2014. [Online]. Available: http://en.wikipedia.org/wiki/AngularJS.

[3] J. Gruber, "Daring Firewall: Markdown," [Online]. Available: http://daringfireball.net/projects/markdown/. [Accessed 2014].

[4] Sensio Labs, "Twig - The flexible, fast, and secure template engine for PHP," Sensio Labs, [Online]. Available: http://twig.sensiolabs.org/. [Accessed 2014].

[5] T. Berners-Lee, "Information Management: A Proposal," 1989. [Online]. Available: http://www.w3.org/History/1989/proposal-msw.html.

[6] T. Berners-Lee, R. Cailliau, A. Luotonen, H. Frystyk Nielsen and A. Secret, "The World-Wide Web," *Communications of the ACM*, vol. 37, no. 8, pp. 76-82, 1994.

[7] T. Berners-Lee, J. Hendler and O. Lassila, "The Semantic Web," *Scientific American*, vol. 284, no. 5, pp. 34-43, 2001.

[8] C. Marshall and F. Shipman, "Which semantic web?," in *Proceedings of*

the fourteenth ACM conference on Hypertext and hypermedia, 2003.

[9] T. Berners-Lee, "Linked Data - Design Issues," 27 07 2006. [Online]. Available: http://www.w3.org/DesignIssues/LinkedData.html.

[10] A. Jentzsch, R. Cyganiak and C. Bizer, "State of the LOD Cloud," 19 09 2011. [Online]. Available: http://lod-cloud.net/state/. [Accessed 2014].

[11] T. Heath and C. Bizer, Linked Data: Evolving the Web into a Global Data Space, Morgan & Claypool, 2011.

[12] World Wide Web Consortium (W3C), "Multimedia Semantics: Overview of Relevant Tools and Resources," [Online]. Available: http://www.w3.org/2005/Incubator/mmsem/wiki/Tools_and_Resources. [Accessed 2014].

[13] S. Taylor, N. Jekjantuk, M. Chris and J. Z. Pan, "dot.rural Digital Economy Hub," [Online]. Available: http://www.dotrural.ac.uk/curios/uploads/TJMP2013.pdf.

[14] World Wide Web Consortium (W3C), "WORKSHOP REPORT: LINKED ENTERPRISE DATA PATTERNS WORKSHOP," 6-7 December 2011. [Online]. Available: http://www.w3.org/2011/09/LinkedData/Report.

[15] W3C Web Applications (WebApps) Working Group, "WebComponents," 2014. [Online]. Available: http://www.w3.org/wiki/WebComponents/.

[16] Wikimedia Foundation, Inc., "Model–view–controller," 2014. [Online]. Available: http://en.wikipedia.org/wiki/Model-view-controller.

[17] J. Rayfield, P. Wilton, T. Grahame and S. Williams, "BBC Ontologies - Sports Ontology," BBC, [Online]. Available: http://www.bbc.co.uk/ontologies/sport. [Accessed 2014].

[18] E. Giralt, "SkyData: Linked Open Data Framework (PHP OOP + RDF + Angular + NoSQL)," GitHub, Inc, 2014. [Online]. Available: https://github.com/egiralt/skydata.

[19] M. Schneider and G. Sutcliffe, "Reasoning in the OWL 2 Full Ontology Language using First-Order Automated Theorem Proving," *Proc. CADE 23*, vol. 6803, pp. 446-460, 2011.

[20] T. Heath, "Linked Data - Connect Distributed Data across the Web," [Online]. Available: http://linkeddata.org/. [Accessed 2014].

[21] K. Alexander, M. Hausenblas, R. Cyganiak and J. Zhao, "Describing Linked Datasets - On the Design and Usage of voiD, the 'Vocabulary of Interlinked Datasets," in *WWW 2009 Workshop: Linked Data on the Web LDOW2009*, Madrid, 2009.

[22] Wikipedia, "Content management system," [Online]. Available: http://en.wikipedia.org/wiki/Content_management_system.

[23] "Apache Jena," Apache Software Foundation, 2011-2014. [Online]. Available: https://jena.apache.org/.

[24] T. Berners-Lee, "WWW Past & Future - Berners-Lee - Royal Society - slide "Future: Stack of expressive power"," 2003. [Online]. Available: http://www.w3.org/2003/Talks/0922-rsoc-tbl/slide30-0.html.