German Medical Data Sciences: Visions and Bridges
R. Röhrig et al. (Eds.)

doi:10.3233/978-1-61499-808-2-170

Light-Weighted Automatic Import of Standardized Ontologies into the Content Management System Drupal

Christoph BEGER [a,b,1], Alexandr UCITELI [a] and Heinrich HERRE [a]
[a] *University of Leipzig, Institute for Medical Informatics, Statistics and Epidemiology*
[b] *University of Leipzig, University Hospital, Wachstumsnetzwerk CrescNet*

Abstract. The amount of ontologies, which are utilizable for widespread domains, is growing steadily. BioPortal alone, embraces over 500 published ontologies with nearly 8 million classes. In contrast, the vast informative content of these ontologies is only directly intelligible by experts. To overcome this deficiency it could be possible to represent ontologies as web portals, which does not require knowledge about ontologies and their semantics, but still carries as much information as possible to the end-user. Furthermore, the conception of a complex web portal is a sophisticated process. Many entities must be analyzed and linked to existing terminologies. Ontologies are a decent solution for gathering and storing this complex data and dependencies. Hence, automated imports of ontologies into web portals could support both mentioned scenarios. The Content Management System (CMS) Drupal 8 is one of many solutions to develop web presentations with less required knowledge about programming languages and it is suitable to represent ontological entities. We developed the Drupal Upper Ontology (DUO), which models concepts of Drupal's architecture, such as nodes, vocabularies and links. DUO can be imported into ontologies to map their entities to Drupal's concepts. Because of Drupal's lack of import capabilities, we implemented the Simple Ontology Loader in Drupal (SOLID), a Drupal 8 module, which allows Drupal administrators to import ontologies based on DUO. Our module generates content in Drupal from existing ontologies and makes it accessible by the general public. Moreover Drupal offers a tagging system which may be amplified with multiple standardized and established terminologies by importing them with SOLID. Our Drupal module shows that ontologies can be used to model content of a CMS and vice versa CMS are suitable to represent ontologies in a user-friendly way. Ontological entities are presented to the user as discrete pages with all appropriate properties, links and tags.

Keywords. Automatic data processing, biomedical ontology, information storage and retrieval, knowledge bases, metadata

1. Introduction

Drupal is an open source Content Management System (CMS). It was initially release in 2001 and ranks on the third place of all CMS ordered by market share with approximately 5%, according to Web Technology Surveys [1]. The benefit of using a CMS like Drupal is that users do not need any programming skills to create content on CMS based webpages. Drupal in its initial form, allows the user to create content as "node" with a

[1] Corresponding Author: **Christoph.Beger@imise.uni-leipzig.de**

comprehensive User Interface. Concrete nodes can be tagged by predefined vocabulary terms and the terms allow users to search for content with specific common features. Drupal profits from an active community, which develops many new functionalities via modules. Those modules can be installed in Drupal to enrich its features.

Leipzig Health Atlas (LHA) is an ongoing project at the University of Leipzig, funded by the German Ministry of Education and Research (reference number: 031L0026, program: i:DSem – Integrative Datensemantik in der Systemmedizin). The goal of LHA is to integrate information about finished projects, publications and corresponding data sets on a website to allow interested persons access of metadata and to possibly support upcoming scientific projects. In LHA we developed an ontology to describe mentioned entities, their properties and complex ties. After this step we uncovered the demand for a module which is capable of importing ontologies into Drupal.

At the time of writing this article there exists no reliable method to import ontologies into Drupal 8, which preserves hierarchies and properties of entities. Indeed, there is a Drupal module called Open Semantic Framework [2], but it is only available for version 7 and it does not incorporate ontological entities in Drupal's default database. Therefore if one wants to transfer data from an ontology into Drupal 8, it has to be done the manual way by creating content by hand. This deficiency is acceptable for small ontologies, but with growing size the required time increases to an unreasonable state.

The implementation of a Drupal module could simplify imports of ontologies significantly. Also, importing ontologies in Drupal could make them accessible to users without knowledge about ontologies, while motivating developers to model their content in ontologies, aware that those ontologies are easy to import in Drupal. Further, already existing and established ontologies/terminologies could be used to automatically create vocabularies in Drupal by extracting all required classes.

2. Methods

Based on the "Three ontology method" [3] we developed the Drupal Upper Ontology (DUO) as Task Ontology and integrated our LHA Ontology as Domain Ontology into it. DUO models concepts of Drupal's architecture, such as nodes, vocabularies and links.

2.1. The Drupal Upper Ontology

We developed an ontology which represents default components of Drupal (fields, nodes, files, vocabularies). The idea is to provide users predefined Drupal concepts which shall be specialized and instantiated in a domain ontology, so that the import module is able to map entities of the domain ontology into Drupal's database. Note that Drupal's fields (e.g. *title* or *content*) are represented in DUO as properties, node types (e.g. *Project* or *Publication*) and vocabulary tags (e.g. *Disease1*) as classes and concrete nodes (e.g. *Project1*) as individuals. Additional entities or files may be instantiated for reusability.

Figure 1 shows the structure of DUO and an exemplary Domain Ontology. The dashed connections between both ontologies symbolize instantiation, subclass and subproperty relations respectively. For integration and formal foundation of both, Task and Domain Ontology [3], we used the General Formal Ontology (GFO) [4] as Top-Level Ontology. In GFO sense we distinguish between symbolic structures (content of the pages, like text and images) and the entities (concepts or individuals, like concrete projects), that are represented by the symbolic structures. For the sake of simplicity we

consider *duo:Entity*, *duo:Node* and *duo:File* as subclasses of *gfo:Individual* and *duo:Vocabulary* as a subclass of *gfo:Concept*. The precise ontological analysis of DUO entities is out of scope of this paper and will be covered in a separate publication.

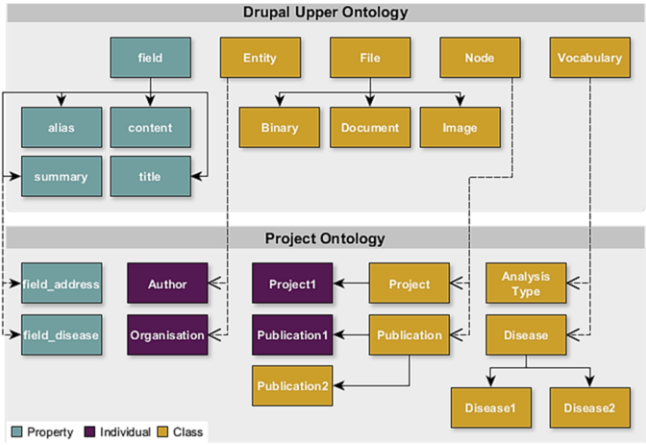

Figure 1. Structure of DUO as ER diagram with connections to a generic Domain Ontology.

2.2. Guideline for Ontology Conversion

In this section we explain how to modifying an existing or developing a new ontology to make it importable with our module. To avoid errors during the import process we defined some restrictions and requirements for an importable ontology. An ontology must import DUO, because only classes and properties which are defined by DUO and specialized or instantiated by the importing ontology itself are recognized. All remaining entities are ignored. Properties must match machine-readable field names in Drupal.

The next step concerns classifications of individuals (nodes) by concepts of vocabularies. To make the class hierarchies available as vocabularies you have to define a new subclass of *duo:Vocabulary* and assign all affected root classes to it. This allows the module to recognize vocabularies and their hierarchies.

Drupal is based on publishing pages with specific content. To denote individuals, which should be represented as page, assign them to a subclass of *duo:Node*. The subclass corresponds to Drupal's content type. It is also possible to import classes as Drupal pages. Additionally data respective annotation properties specify the values of node fields and object respective annotation properties denote links between node pages.

3. Results

We implemented a Drupal 8 compatible module[2], which is able to parse OWL or JSON files and imports extracted data by using Drupal's inbuilt API. Our module compensates Drupal's lack of an ontology import. Most ontologies can be easily transformed with small adjustments and resulting normalized ontologies are imported within a few steps.

[2] The module is available as download from the releases section of our GitHub repository: *https://github.com/Onto-Med/SOLID/releases*

The creation of the LHA metadata ontology was an ongoing process but very early iterations already consisted of approximately 100 classes, 200 individuals and 2000 axioms. Within a few minutes and with hardly any effort we imported mentioned metadata and implemented an automated pipeline, which periodically synchronized Drupal's database records and the ontology by using our module.

The Simple Ontology Loader in Drupal (SOLID) consists of components for file handling/parsing and data import. A file handler is used to parse uploaded files and transform the content into a PHP object. Depending on the file format, a factory serves the appropriate file handler. Additional file handlers can be added by inheriting *AbstractFileHandler*. At its current state the module implements two handlers, one for OWL (using the integrated EasyRdf library [5] of Drupal 8) and one for JSON. The JSON file is directly convertible to a PHP object and thus does not require conversions.

The file handler for OWL files extracts all classes beneath *duo:Vocabulary*. Direct children denote separate taxonomies and all children of these classes are hierarchical tags. In contrast to vocabularies, nodes can be individuals or classes, which instantiate direct subclasses of *duo:Node* or which are subclasses of them respectively. Their properties are stored as data, object or annotation properties. To extract the properties for an entity, the module iterates over all defined properties of the ontology. Imported properties are ignored, unless they are part of DUO. The module automatically resolves references to other individuals and extracts data depending on the instantiated class of the individual.

4. Discussion

Our module allows the import of any ontology (which imports DUO) without much effort. We expect that this novel way of presenting ontologies within Drupal will lower the barrier to access semantical structures, because ontological Entities and their relations are depicted with traditional websites and links. Furthermore the module enables ontology experts to model complex content of web portals as ontology and import them into Drupal.

Different approaches exist to combine semantics with CMS. For example Das et al. [6] introduced the Science Collaboration Framework (SCF), which supports web communities of the biomedical domain with semantical awareness. The authors describe a Drupal-based architecture, which can be enriched by gene ontologies. They mention proxies for referencing genes from multiple RDF-repositories, so that Drupal content is annotatable with those genes. Proxies are not generic, hence require implementations for each data source. The SCF enables annotation of existing Drupal content with ontology concepts, whereas SOLID allows to import semantically enriched content directly into Drupal so that its structures and properties are directly accessible by domain experts.

There are other domains where the amount of available data increases. For example according to [7] the recent "improvements in DNA sequencing have vastly increased the quantity and availability of genomic […] data". Therefore, the authors of mentioned article developed the construction toolkit Tripal, which is a combination of Drupal and Chado, a database schema for storage of biological data. Where our import module transforms datasets into Drupal entities, Tripal brings a new database schema to store datasets and provides management, visualization and search capabilities. Both approaches are important in terms of data accessibility, but Tripal is more appropriate for domain experts for genomic data, where our module aims to present more general ontological data to non-domain experts.

Above mentioned articles are Drupal-based. Befa et al. [8] details an extension for the CMS Alfresco [9], which facilitates import and export of content as RDF/S ontology. The extension only supports data properties, whereas our module understands OWL syntax and allows the import of all property types. If a proper use case exists for Drupal content export as OWL, we may enhance our module with an export functionality.

In future, more and more sources of data will emerge and tools like SOLID will be necessary to improve the accessibility of highly domain specific data, which is also utilizable for other research areas.

5. Conclusion

Ontologies are available in a vast number to many domains, but they suffer from poor accessibility for non-ontologists. We developed a novel approach to overcome mentioned deficiency, by implementing a Drupal 8 module with import capabilities for ontologies. Our module enables an expert to present his ontologies to the general public, because ontological entities are transformed into Drupal nodes. Moreover, the module enhances the conception of a web portal, because complex entities and their relations can be easily modeled with an ontology. In consequence, the entities are importable with our module. All functionalities of DUO and SOLID were successfully validated with selected datasets of LHA. In time of writing this article, the design of the LHA Web Portal is not finished and thus is not available for public use.

6. Conflict of Interest

The authors state that they have no conflict of interests.

References

[1] W³ Techs, *Usage Statistics and Market Share of Content Management Systems for Websites, February 2017*, Q-Success, 2017. [Online]. Available: https://w3techs.com/technologies/overview/content_management/all. [Accessed 10 02 2017].

[2] Structured Dynamics LLC, *Open Semantic Framework*, Structured Dynamics LLC, 2017. [Online]. Available: http://opensemanticframework.org. [Accessed 10 02 2017].

[3] R. Hoehendorf, A.-C. N. Ngomo and H. Herre, *Developing Consistent and Modular Software Models with Ontologies*, Proceedings of the Eighth SoMeT_09, pp. 399-412, 2009.

[4] H. Herre, B. Heller, P. Burek, R. Hoehendorf, F. Loebe and H. Michalek, *General Formal Ontology (GFO) – A foundational ontology integrating objects and processes [Version 1.0.1]*, Leipzig, University of Leipzig, IMISE, 2007.

[5] N. Humfrey, *EasyRdf - RDF Library for PHP*, Available: http://www.easyrdf.org. [Accessed 10 02 2017].

[6] D. Sudeshna, L. Girard, T. Green, L. Weitzman, A. Lewis-Bowen and T. Clark, *Building biomedical web communities using a semantically aware content management system*, Brief Bioinform, pp. 129-138, December 2008.

[7] S. P. Ficklin, L.-A. Sanderson, C.-H. Cheng, M. E. Staton, T. Lee, I.-H. Cho, S. Jung, K. E. Bett and D. Main, *Tripal: a construction toolkit for online genome databases*, Database (Oxford), 29 Semptember 2011.

[8] M. Befa, E. Kontopoulos, N. Bassiliades, C. Berberidis and I. Vlahavas, *Deploying a Semantically-Enabled Content Management System in a State University*, Electronic Government and the Information Systems Perspective, EGOVIS 2010, Lecture Notes in Computer Science, no. 6267, 2010.

[9] Alfresco Software, Inc., *Activate Process and Content to Make Business Flow | Alfresco*, 2017. [Online]. Available: https://www.alfresco.com. [Accessed 8 May 2017].