Using Semantic Programming for Developing a Web Content Management System for Semantic Phenotype Data

Vogt, Lars[1[0000-0002-8280-0487]]✉; Baum, Roman[1[0000-0001-5246-9351]]; Köhler, Christian[1[0000-0001-6966-7901]]; Meid, Sandra[1[0000-0003-4627-1853]]; Quast, Björn[2[000-0002-3760-5834]] and Grobe, Peter[2[0000-0003-4991-5781]]

[1] Universität Bonn, IEZ, An der Immenburg 1, 53121 Bonn, Germany
[2] Zoologisches Forschungsmuseum Alexander Koenig, Adenauerallee 160, 53113 Bonn, Germany
lars.m.vogt@googlemail.com

Abstract. We present a prototype of a semantic version of Morph·D·Base that is currently in development. It is based on SOCCOMAS, a semantic web content management system that is controlled by a set of source code ontologies together with a Java-based middleware and our Semantic Programming Ontology (SPrO). The middleware interprets the descriptions contained in the source code ontologies and dynamically decodes and executes them to produce the prototype. The Morph·D·Base prototype in turn allows the generation of instance-based semantic morphological descriptions through completing input forms. User input to these forms generates data in form of semantic graphs. We show with examples how the prototype has been described in the source code ontologies using SPrO and demonstrate live how the middleware interprets these descriptions and dynamically produces the application.

Keywords: semantic programming, phenotypic data, linked open data, semantic Morph·D·Base, semantic annotation, morphological data

1 Introduction

Ontologies are dictionaries that consist of labeled classes with definitions that are formulated in a highly formalized canonical syntax and standardized format (e.g. Web Ontology Language, OWL, serialized to the Resource Description Framework, RDF), with the goal to yield a lexical or taxonomic framework for knowledge representation [1]. Ontologies are often formulated in OWL and thus can be documented in the form of class-based semantic graphs[1]. Ontologies contain commonly accepted domain

[1] A semantic graph is a network of RDF/OWL-based triple statements, in which a given Uniform Resource Identifier (URI) takes the *Object* position in one triple and the *Subject* position in another triple. This way, several triples can be connected to form a semantic graph. Because information about individuals can be represented as a semantic graph as well, we distinguish class- and instance-based semantic graphs.

knowledge about specific kinds of entities and their properties and relations in form of classes defined through universal statements [2,3], with each class possessing its own URI, through which it can be identified and individually referenced. Ontologies in this sense do not include statements about individual entities. Statements about individual entities are assertional statements. In an assertional statement individuals can be referred to through their own URI and their class affiliation can be specified by referencing this class' URI. If assertional statements are grounded in empirical knowledge that is based on observation and experimentation, we refer to them as empirical data. Empirical data can be formulated in OWL and thus documented in the form of instance-based semantic graphs. As a consequence, not every OWL file and not every semantic graph is an ontology—it is an ontology only if it limits itself to express universal statements about kinds of entities [3]. A knowledge base, in contrast, consists of a set of ontology classes that are populated with individuals and assertional statements about these individuals [3] (i.e. data). Ontologies do not represent knowledge bases, but are part of them and provide a means to structure them [4].

By providing a URI for each of their class resources, ontologies can be used to substantially increase semantic transparency and computer-parsability for all kinds of information. Respective URIs are commonly used for semantically enriching documents and annotating database contents to improve integration and interoperability of data, which is much needed in the age of Big Data, Linked-Open-Data and eScience [5–7]. Ontologies and their URIs also play an important role in making data maximally findable, accessible, interoperable and reusable (see FAIR guiding principle [8]) and in establishing eScience-compliant (meta)data standards [6,7,9–12].

An increasing number of organizations and institutions recognize the need to comply with the FAIR guiding principle and seek for technical solutions for efficiently managing the accessibility, usability, disseminability, integrity and security of their data. Content management systems in form of knowledge bases (i.e. Semantic web content management systems, S-WCMS) have the potential to provide a solution that meets both the requirements of organizations and institutions as well as of eScience.

Despite the obvious potential of ontologies and semantic technology in data and knowledge management, their application is usually restricted to annotating existing data in relational database applications. Although tuple stores that store information as RDF triple statements are capable of handling large volumes of triples and although semantic technology facilitates detailed data retrieval of RDF/OWL-based data through SPARQL [13] endpoints and inferencing over OWL-based data through semantic reasoners, not many content management systems have implemented ontologies to their full potential. We believe that this discrepancy can be explained by a lack of application development frameworks that are well integrated with RDF/OWL.

2 Semantic Programming

2.1 Semantic Programming Ontology (SPrO)

With SPrO [14] we extend the application of ontologies from providing URIs for annotating (meta)data and documenting data in form of semantic graphs stored and

managed in a S-WCMS to using an ontology for software programming. We use SPrO like a programming language with which one can control a S-WCMS by describing it within a corresponding source code ontology. SPrO defines ontology resources in the form of classes, individuals and properties that the accompanying Java-based middleware interprets as a set of commands and variables. The commands are defined as annotation properties. Specific values and variable-carrying resources are defined as ontology individuals. Additional object properties are used to specify relations between resources, and data properties are used for specifying numerical values or literals for resources that describe the S-WCMS.

SPrO can be used to describe all features, workflows, database processes and functionalities of a particular S-WCMS, including its graphical user interface (GUI). The descriptions at their turn are contained in one or several source code ontologies in form of annotations of ontology classes and ontology individuals. Each annotation consists of a command followed by a value, index or resource and can be extended by axiom annotations and, in case of individuals, also property annotations. Contrary to other development frameworks that utilize ontologies (e.g. [15,16]), you can use the resources of SPrO to describe a particular content management application within its corresponding source code ontology. The application is thus self-describing. The accompanying Java-based middleware decodes the descriptions as declarative specifications of the content management application, interprets them and dynamically executes them on the fly. We call this approach semantic programming.

2.2 Semantic Ontology-Controlled Application for Web Content Management Systems (SOCCOMAS)

SOCCOMAS [17] is a semantic web content management system that utilizes SPrO and its associated middleware. It consists of a basic source code ontology for SOCCOMAS itself (SC-Basic), which contains descriptions of features and workflows typically required by a S-WCMS, such as user administration with login and signup forms, user registration and login process, session management and user profiles, but also publication life-cycle processes for data entries (i.e. collections of assertional statements referring to a particular entity of a specific kind, like for instance a specimen) and automatic procedures for tracking user contributions, provenance and logging change-history for each editing step of any given version of a data entry. All data and metadata are recorded in RDF following established (meta)data standards using terms and their corresponding URIs from existing ontologies. Each S-WCMS run by SOCCOMAS provides human-readable output in form of HTML and CSS for browser requests and access to a SPARQL endpoint for machine-readable service requests. Moreover, it assigns a DOI to each published data entry and data entries are published under a creative commons license. When a data entry is published, it becomes openly and freely accessible through the Web. Hence, all data published by a S-WCMS run by SOCCOMAS reaches the five star rank of Tim Berners-Lee's rating system for Linked Open Data [18].

The descriptions of the features, processes, data views, HTML templates for input forms, specifications of input control and overall behavior of each input field of a particular S-WCMS are contained in its accompanying source code ontology, which

is specifically customized to the needs of that particular S-WCMS. These descriptions also include specifications of the underlying data scheme that determines how user input triggers the generation of data-scheme-compliant triple statements and where these triples must be saved in the Jena tuple store in terms of named graph[2] and workspace (i.e. directory). For instance the morphological data repository semantic Morph·D·Base has its own source code ontology for its morphological description module (SC-MDB-MD [20]) that is specifically customized to the needs of semantic Morph·D·Base [19] (Fig. 1).

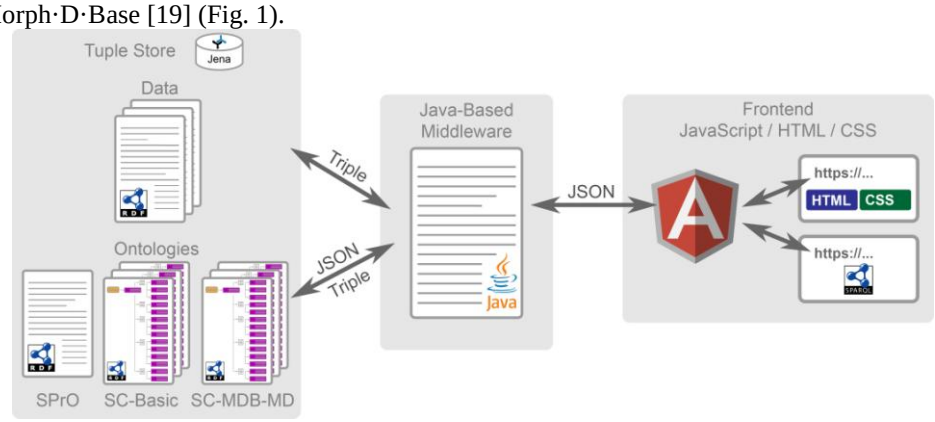

Fig. 1. Overall workflow of semantic Morph·D·Base [19] run by SOCCOMAS. Left: Jena tuple store containing the data of semantic Morph·D·Base as well as (i) the Semantic Programming Ontology (SPrO), which contains the commands, subcommands and variables used for describing semantic Morph·D·Base, (ii) the source code ontology for SOCCOMAS (SC-Basic), which contains the descriptions of general workflows and features that can be used by any S-WCMS, and (iii) the particular source code ontology for the morphological description module of semantic Morph·D·Base (SC-MDB-MD), which has been individually customized to contain the description of all features that are special to semantic Morph·D·Base. Middle: the Java-based middleware. Right: the frontend based on the JavaScript framework AngularJS with HTML and CSS output for browser requests and access to a SPARQL endpoint for machine requests.

This way, the developers of semantic Morph·D·Base can use the general functionality that comes with SC-Basic and add upon that the features specifically required for semantic Morph·D·Base by describing them in SC-MDB-MD using the commands, values and variable-carrying resources from SPrO. After semantic Morph·D·Base goes online, its developers can still describe new input fields in SC-MDB-MD or new types of data entries in respective additional source code ontologies and therewith update semantic Morph·D·Base without having to program in other layers.

The application descriptions contained in SC-Basic and SC-MDB-MD organize the Jena tuple store into different workspaces, which at their turn are organized into dif-

[2] A named graph identifies a set of triple statements by adding the URI of the named graph to each triple belonging to this named graph, thus turning the triple into a quad. The Jena tuple store can handle such quadruples. The use of named graphs enables partitioning data in an RDF store.

ferent named graphs, each of which belongs to a particular class of named graphs. This enables differentially storing data belonging to a specific entry or version of an entry into different named graphs, which in turn allows for flexible and meaningful fragmentation of data and flexible definition of different data views.

2.3 Semantic Morph·D·Base as a Use-Case

Morphological data drive much of the research in life sciences [21,22], but are usually still published as morphological descriptions in form of unstructured texts, which are not machine-parsable and often hidden behind a pay-wall. This not only impedes the overall findability and accessibility of morphological data. Due to the immanent semantic ambiguity of morphological terminology, researchers who are not experts of the described taxon will have substantial problems comprehending and interpreting the morphological descriptions (see *Linguistic Problem of Morphology* [23]). This semantic ambiguity substantially limits the interoperability and reusability of morphological data, with the consequence that morphological data usually do not comply with the FAIR guiding principles [8].

Semantic Morph·D·Base [19] enables users to generate highly standardized and formalized morphological descriptions in the form of assertional statements represented as instance-based semantic graphs. The main organizational backbone of a morphological description is a partonomy of all the anatomical parts and their subparts of the specimen the user wants to describe. Each such part possesses its own URI and is indicated to be an instance of a specific ontology class. Semantic Morph·D·Base allows reference to ontology classes from all anatomy ontologies available at BioPortal [24]. Parts can be further described (i) semantically through defined input forms, often referencing specific ontology classes from PATO [25], resulting in an instance-based semantic graph that we call a *Semantic Instance Anatomy* [26,27], (ii) as semantically enriched free text, and (iii) through images with specified regions of interest which can be semantically annotated. All this information is stored in the tuple store and can be accessed through a web-based interface and a SPARQL endpoint. The *Semantic Instance Anatomy* graph is meaningfully fragmented into a sophisticated scheme of named graph resources, which additionally supports subsequent data retrieval and data analyses.

Because semantic Morph·D·Base is run by SOCCOMAS, all description entries not only possess their own unique URI, but also receive their own DOI when they are published and are freely and openly accessible through the Web. All data and metadata are stored as RDF triples in a Jena tuple store and can be searched using a SPARQL endpoint. Instances and classes referenced in these triples have their own globally unique and persistent identifiers and are findable through the endpoint. Both metadata as well as the descriptions themselves reference resources of well established ontologies, which substantially increases their interoperability and reusability. As a consequence, data and metadata in semantic Morph·D·Base comply with the FAIR principles.

Link to a live-demo of semantic Morph·D·Base: https://proto.morphdbase.de/

References

1. Smith, B.: Ontology. In: Floridi, L. (ed.) Blackwell Guide to the Philosophy of Computing and Information, pp. 155–166. Blackwell Publishing, Oxford (2003).
2. Schulz, S., Stenzhorn, H., Boeker, M., Smith, B.: Strengths and limitations of formal ontologies in the biomedical domain. RECIIS **3**, 31–45 (2009).
3. Schulz, S., Jansen, L.: Formal ontologies in biomedical knowledge representation. IMIA Yearb Med informatics 2013 **8**, 132–146 (2013).
4. Uschold, M., Gruninger, M.: Ontologies: Principles, Methods and Applications. Knowl Eng Rev **11**, 39–136 (1996).
5. Sansone, S.-A., Rocca-Serra, P., Tong, W., Fostel, J., Morrison, N., et al.; A Strategy Capitalizing on Synergies: The Reporting Structure for Biological Investigation (RSBI) Working Group. Omi A J Integr Biol **10**, 164–171 (2006).
6. Vogt, L.: The future role of bio-ontologies for developing a general data standard in biology: chance and challenge for zoo-morphology. Zoomorphology **128**, 201–217 (2009).
7. Vogt, L., Nickel, M., Jenner, R.A., Deans, A.R.: The Need for Data Standards in Zoomorphology. J Morphol **274**, 793–808 (2013).
8. Wilkinson, M.D., Dumontier, M., Aalbersberg, I.J., Appleton, G., Axton, M., et al.: The FAIR Guiding Principles for scientific data management and stewardship. Sci Data **3**, 160018 (2016).
9. Brazma, A.: 2001) On the importance of standardisation in life sciences. Bioinformatics **17**, 113–114 (2001).
10. Brazma, A., Hingamp, P., Quackenbush, J., Sherlock, G., Spellman, P., et al.: Minimum information about a microarray experiment (MIAME)–toward standards for microarray data. Nat Genet **29**, 365–371 (2001).
11. Wang, X., Gorlitsky, R., Almeida, J.S.: From XML to RDF: how semantic web technologies will change the design of "omic" standards. Nat Biotechnol **23**, 1099–1103 (2005).
12. Vogt, L.: eScience and the need for data standards in the life sciences: in pursuit of objectivity rather than truth. Syst Biodivers **11**, 257–270 (2013).
13. SPARQL Query Language for RDF. W3C Recommendation 15 January 2008.
14. GitHub: Code for Semantic Programming Ontology (SPrO). Available: https://github.com/SemanticProgramming/SPrO.
15. Wenzel, K.: KOMMA : An Application Framework for Ontology-based Software Systems. Semant Web J **swj89_0**: 1–10 (2010).
16. Buranarach, M., Supnithi, T., Thein, Y.M., Ruangrajitpakorn, T., Rattanasawad, T., et al.: OAM: An Ontology Application Management Framework for Simplifying Ontology-Based Semantic Web Application Development. Int J Softw Eng Knowl Eng **26**, 115–145 (2016).
17. GitHub: Code for Semantic Ontology-Controlled Web Content Management System (SOCCOMAS). Available: https://github.com/SemanticProgramming/SOCCOMAS.
18. Berners-Lee, T.: Linked Data. (2009) Available: https://www.w3.org/DesignIssues/LinkedData.html.
19. Semantic Morph•D•Base Prototype. Available: https://proto.morphdbase.de.
20. GitHub: Code for semantic Morph·D·Base prototype. Available: https://github.com/MorphDBase/MDB-prototype.
21. Deans, A.R., Lewis, S.E., Huala, E., Anzaldo, S.S., Ashburner, M., et al.: Finding Our Way through Phenotypes. PLoS Biol **13**, e1002033 (2015).

22. Mikó, I., Deans, A.R.: Phenotypes in insect biodiversity research Phenotype data : past and present. In: Foottit, R.G., Adler, P.H. (eds.) Insect Biodiversity: Science and Society, pp. 789–800. John Wiley & Sons, Ltd, Vol. II. (2018).
23. Vogt, L., Bartolomaeus, T., Giribet, G.: The linguistic problem of morphology: structure versus homology and the standardization of morphological data. Cladistics **26**, 301–325 (2010).
24. BioPortal. Available: http://bioportal.bioontology.org/.
25. Phenotype And Trait Ontology (PATO). Available: http://obofoundry.org/ontology/pato.html.
26. Vogt, L.: Assessing similarity: on homology, characters and the need for a semantic approach to non-evolutionary comparative homology. Cladistics **33**, 513–539 (2017).
27. Vogt, L.: Towards a semantic approach to numerical tree inference in phylogenetics. Cladistics **34**, 200–224 (2018).