MODL: A Modular Ontology Design Library

Cogan Shimizu, Quinn Hirt, and Pascal Hitzler

Data Semantics Laboratory, Wright State University, Dayton, OH, USA

Abstract. Pattern-based, modular ontologies have several beneficial properties that lend themselves to FAIR data practices, especially as it pertains to Interoperability and Reusability. However, developing such ontologies has a high upfront cost, e.g. reusing a pattern is predicated upon being aware of its existence in the first place. Thus, to help overcome these barriers, we have developed MODL: a modular ontology design library. MODL is a curated collection of well-documented ontology design patterns, drawn from a wide variety of interdisciplinary use-cases. In this paper we present MODL as a resource, discuss its use, and provide some examples of its contents.

1 Introduction

The Information Age is an apt description for these modern times; between the World Wide Web and the Internet of Things an unfathomable amount of information is accessible to humans and machines, but the sheer volume and heterogeneity of the data have their drawbacks. Humans have difficulty drawing *meaning* from large amounts of data. Machines can parse the data, but do not *understand* it. Thus, in order to bridge this gap, data would need to be organized in such a way that some critical part of the human conceptualization is preserved. Ontologies are a natural fit for this role, as they may act as a vehicle for the sharing of *understanding* [7].

Unfortunately, published ontologies have infrequently lived up to such a promise, hence the recent emphasis on FAIR (Findable, Accessible, Interoperable, and Reusable) data practices [26]. More specifically, many ontologies are not interoperable or reusable. This is usually due to incompatible ontological commitments: strong—or very weak—ontological committments lead to an ontology that is really only useful for a specific use-case, or to an ambiguous model that is almost meaningless by itself.

To combat this, we have developed a methodology for developing so-called modular ontologies [15]. In particular, we are especially interested in pattern-based modules [12]. A modularized ontology is an ontology that individual users can easily adapt to their own use-cases, while still preserving relations with other versions of the ontology; that is, keeping it *interoperable* with other ontologies. Such ontologies may be so adapted due to their "plug-and-play" nature; that is, one module may be swapped out for another developed from the same pattern.

An ontology design pattern is, essentially, a small self-contained ontology that addresses a general problem that has been observed to be invariant over different

domains or applications [10]. By tailoring a pattern to a more specific use-case, an ontology engineer has developed a *module*. This modelling paradigm moves much of the cost away from the formalization of a conceptualization (i.e. the logical axiomatization). Instead, pattern-based modular ontolody design (PBMOD) is predicated upon knowledge of available patterns, as well as being aware of the use-cases it addresses and its ontological commitments.

Thus, in order to address the findability and accessibility aspects of PB-MOD, we have developed MODL: a modular ontology design library. MODL is a curated collection of well-documented ontology design patterns. The particular research contribution is both the curation and documentation. Some of the patterns are novel, but many more have been extracted from existing ontologies and streamlined for use in a general manner. MODL, as an artefact, is distributed online as a collection of annotated OWL files and a technical report containing schema diagrams and explanations of each OWL axiom.[1]

The rest of the paper is organized as follows. Section 2 discusses the relevance of this work. Section 3 presents our Modular Ontology Design Library in detail and in Section 4 we conclude and discuss future work.

2 Relevance

This section provides an overview of where this resource fits into the state of the art, who is expected to use this resource, justification for why we believe that it will be used, and how we plan to disseminate the resource.

2.1 Research Context

Pattern-based modular ontology development is not a conceptually new idea—instead, it is a continuation of an already established paradigm. Both modularization of ontologies [18] and pattern-based modelling [6] have been identified as improvements to the ontology engineering process. These concepts have culiminated in mature paradigms (e.g. MOM [15, 12] and eXtreme Design [17, 2]), both having been used in large-scale projects (e.g. GeoLink [16] and VALCRI [4]). However, the ontology engineering community, especially those that utilize patterns, have indicated an increased need for better tooling support [9, 3], of which there are two complementary aspects: a dedicated development environment and a critical mass of Ontology Design Patterns (or ODPs for short).[2]

There is already a prototype that begins to address the need for a dedicated development environment for pattern-based ontologies [8]. It also provides a set of hard-coded patterns that were extracted (at the time of development) from the ODP Portal.[3] However, having the pattern library tightly coupled with the

---

[1] https://dase.cs.wright.edu/content/modl-modular-ontology-design-library

[2] Anecdotally, one of the more pervasive themes at both the 2018 and 2019 United States Semantic Technologies Symposia (https://us2ts.org/) was a call from ontology engineers in both academia and industry for better tooling support.

[3] http://ontologydesignpatterns.org/wiki/Submissions:ContentOPs

tool is disadvantageous for future development. Indeed, decoupling the tool is desirable, for a number of reasons, as follows.

- Remove the onus of pattern development and upkeep off of the tool developer.
- Enable community driven improvements and tailoring of the library to the end-users use-cases.
- Enable plug-and-play pattern libraries for different domains, etc.

On the other hand, MODL also addresses the crucial need for a critical mass of ODPs. One may argue that this mass exists in the form of the ODP portal. Unfortunately, though, it has suffered under the weight of its own mission. Community enforced quality control has not succeeded in providing a ready-to-use suite of quality patterns for use across multiple domains.

Furthermore, while the quality of a set of patterns is largely subjective, MODL strives for consistency in documentation, uses best practices [14, 11], and limited ontological commitments. In some cases this required polishing extant documentation, writing it from scratch, and tweaking or detecting errors in the formalization. We also include all new schema diagrams [13] following a single paradigm and style.

MODL therefore addresses, in some fashion, both aspects of improving tooling support. In turn, we expect this to lower the barrier of entry to PBMOD, which in turn lowers the barrier of entry for wider adoption semantic web technologies in application areas.

2.2 Adoption & Dissemination

As a new resource, we are certainly interested in reaching a wide audience, ensuring that the resource is visible to the community and adopted into existing workflows. In previous sections we have made the case that MODL is very much a response to a community need and that there is a vibrant, existing community surrounding modular ontology modelling and pattern-based modelling. Additionally, we are currently working very closely with the developer for [8] in order to integrate the MODL modality, i.e. ensure plug-and-play pattern libraries are supported by the dedicated development environment. We discuss concrete next steps in Section 4.

Additionally, we believe that MODL will help bridge communities that utilize a different conceptualization of patterns. For example, the pattern communities surrounding the biological ontologies (e.g. Gene Ontology [1] and the International Conference and Biological Ontology[4]), those communities that utilize OTTR [25], and the corresponding frame semantics found in Framester [5]. At a recent session[5] held during US2TS 2019, bridging the different design pattern communities was discussed; namely, how can future tools incorporate the lessons

---

[4] http://icbo2018.cgrb.oregonstate.edu/node/29
[5] http://us2ts.org/2019/posts/program-session-vii.html

learned from these communities? These next steps are discussed in Sections 3.1 and 4.

Of course, there are the adjacent communities found in the Association for Ontology Design Patterns (ODPA)[6] and the long-standing Workshop on Ontology Design and Patterns,[7] where inspiration for this work originated. Thus, we believe it will be well received. We also wish to continue improving our tutorial on modular ontology modelling, where we think that MODL will be an excellent addition.[8]

Finally, the authors have a number of ongoing collaborations with domain experts in the digital humanities. In terms of adoption and dissemination, this is perhaps our most important connection. Specifically, we are working on building an ontology for representing the records of enslaved people in the historic slave trade.[9] A technical report detailing the modular ontology and its patterns can be found online [23]. There is a lot of excitement surrounding its potential for tying in a myriad of historians' work. Between MODL and [8], we hope to enable an entire community to collaboratively building their ontologies.

3   A Modular Ontology Design Library

In this section, we present in detail MODL. Section 3.1 explains our methodology and the organization of MODL, Section 3.2 provides a brief tutorial on using MODL, Section 3.3 provides an example pattern that has been excerpted from the documentation (some of the language and structure, e.g. subsections, have been adapted to fit this paper format), and finally, Section 3.4 provides information pertaining to accessibility, sustainability, and more.

3.1   MODL's Methodology

MODL is a curated collection of well-documented ontology design patterns. MODL, itself, can be considered to be the combination of two artifacts, the collection of patterns, specified in OWL, and the accompanying documentation. The separation is a little fuzzy, as the OWL serialization is also heavily annotated for convenience. The mission of MODL is to make patterns both findable and accessible. Therefore, it is of utmost importance that every pattern therein is thoroughly documented. One drawback of the ODP Portal is that there are no guidelines provided for documenting the patterns and, during submission, a form is provided with many optional, ill-defined fields. That is not to say all of the patterns documented therein are poorly documented—some patterns did indeed have thorough documentation. Where possible, we preserved these efforts, from

---

[6] `http://ontologydesignpatterns.org/wiki/ODPA`

[7] `http://ontologydesignpatterns.org/wiki/WOP:Main`

[8] `http://ontologydesignpatterns.org/wiki/Training:Tutorial:_Methods_and_Tools_for_Modular_Ontology_Modeling`

[9] `https://enslave.org/`

Category	Patterns
Metapatterns	Explicit Typing
	Property Reification
	Stubs
Organization of Data	Aggregation, Bag, Collection
	Sequence, List
	Tree
Space, Time, and Movement	Spatiotemporal Extent
	Spatial Extent
	Temporal Extent
	Trajectory
	Event
Agents and Roles	AgentRole
	ParticipantRole
	Name Stub
Description and Details	Quantities and Units
	Partonymy/Meronymy
	Provenance
	Identifier

Table 1: This table contains the patterns included in MODL. They have been partitioned into five categories (metapatterns; organization of data; space, time, and movement; agents and roles; and description and details) which are loosely defined by their general use-cases.

either the portal or associated publication, and corresponding credit is given in the MODL documentation.

However, for many of the patterns included in MODL, we needed to fill some gaps. For this we have elected to follow the guidelines set forth in [14]. These guidelines are a result of a community wide survey that ranks the perceived importance of ten different components of ODP documentation. For our purposes, we have chosen to include the top seven. They are *Schema Diagram*, *Example of Pattern Instantiation*, *Compentency Questions*, *Axiomatization*, *OWL File*, *Pointers to Related Patterns*, and *Metadata*. The remaining three components (*Set of Example SPARQL Queries*, *Examples of Available Datasets for Population*, and *Constraints Using ShEx*[10]) are being considered for future versions of MODL.[11]

The schema diagrams for our documentation were manually created using the algorithm found in [13, 21]. We elected to use a simplified visual syntax that conveyed relations between concepts and also contains visual cues for identifying concepts that should be used as 'hooks' into the ODP.

---

[10] http://shex.io/

[11] Furthermore, there is some community indecision on embracing ShEx or SHACL, a newer W3C recommendation. More information can be found at https://www.w3.org/TR/shacl/.

The provided OWL files for each of the patterns are annotated with the Extended Ontology Design Pattern Representation Language (OPLa)[12] [11]. This allows us to embed provenance metadata (e.g. where did this pattern originate?) or provide pointers to related patterns (e.g. generalizations or specializations of the pattern) in annotations. We discuss future extensions to OPLa and how they may be leveraged to improve MODL in Section 4.

Table 1 lists the patterns included in MODL. They have been loosely organized into five categories: metapatterns; organization of data; space, time, and movement; agents and roles; and description and details.

Metapatterns This category contains patterns that can be considered to be "patterns for patterns." In other literature, notably [6], they may be called *structural ontology design patterns*, as they are independent of any specific context, i.e. they are content-independent. This is particularly true for the metapattern for property reification, which, while a modelling strategy, is also a workaround for the lack of *n*-ary relationships in OWL. The other metapatterns address structural design choices frequently encountered when working with domain experts. They present a best practice to non-ontologists for addressing language specific limitations.

Organization of Data This category contains patterns that pertain to how data might be organized. These patterns are necessarily highly abstract, as they are ontological reflections of common data structures in computer science. The pattern for aggregation, bag, or collection is a simple model for connecting many concepts to a single concept. Analogously, for the list and tree pattterns, which aim to capture ordinality and acyclicity, as well. More so than other patterns in this library, these patterns provide an axiomatization as a high-level framework that must be specialized (or modularized) to be truly useful.

Space, Time, and Movement This category contains patterns that model the movement of a thing through a space or spaces and a general event pattern. The semantic trajectory pattern is a more general pattern for modelling the discrete movements along some dimensions. The spatiotemporal extent pattern is a trajectory along the familiar dimensions of time and space. Both patterns are included for convenience.

Agents and Roles This category contains patterns that pertain to agents interacting with things. Here, we consider an agent to be anything that performs some action or role. This is important, as it decouples the role of an agent from the agent itself. For example, a Person may be Husband and Widower at some point, but should not be both simultaneously. These patterns enable the capture of this data. In fact, the agent role and participante role patterns are convenient specializations of property reification that have evolved into a modelling practice writ large. In this category, we also include the name stub, which is a convenient instantiation of the stub metapattern; it allows us to acknowledge that a name is a complicated thing, but sometimes we only really need the string representation.

Description and Details This category contains patterns that model the description of things. These patterns are relatively straightforward, models for

---

capturing "how much?" and "what kind?" for a particular thing; patterns that are derived from Winston's part-whole taxonomy [24]; a pattern extracted from PROV-O [19], perhaps to be used to answer "where did this data come from?"; and a pattern for associating an identifier with something.

3.2 Using MODL

There are two different ways to use MODL—for use in ontology modelling and for use in tools. In both cases, MODL is distributed as a ZIP archive of the patterns' OWL files and accompanying documentation. In the case of the Ontology Engineer, it is simply used as a resource while building an ontology, perhaps by using Modular Ontology Modelling or eXtreme Design methodologies. For the tool developer, we also supply an ontology consisting of exactly the OPLa annotations from each pattern that pertain to OntologicalCollection. As OPLa is fully specified in OWL, these annotations make up an ontology of patterns and their relations. One particular use-case that we foresee is a tool developer querying the ontology for which patterns are related to the current pattern, or looking for a pattern based on keywords or similarity to competency questions.

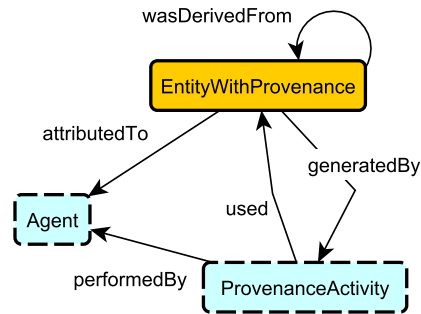## 3.3 Excerpt from Pattern Documentation

EntityWithProvenance

Fig. 1: This figure depicts the schema diagram for the EntityWithProvenance Pattern which is essentially the core of the Provenance Ontology (PROV-O). Yellow boxes are concepts. Light blue boxes with a dashed border are external patterns that the developer may want to also make into a module.

Summary
Figure 1 depicts the schema diagram for the Provenance pattern, as included

in MODL. The EntityWithProvenance Pattern is extracted from the PROV-O ontology. At the pattern level, we do not want to make the ontological committment to a full-blown ontology. It suffices to align a sub-pattern to the core of PROV-O. [19]

The EntityWithProvenance class is any item of interest to which a developer would like to attach provenance information. That is they are interested in capturing, who or what created that item, what was used to derive it, and what method was used to do so. The "who or what" is captured by using the Agent class. The property, wasDerivedFrom is eponymous—it denotes that some set of resources was used during the ProvenanceActivity to generate the EntityWith-Provenance.

Axiomatization[13]

$$\exists \text{attributedTo.Agent} \sqsubseteq \text{EntityWithProvenance} \tag{1}$$

$$\text{EntityWithProvenance} \sqsubseteq \forall \text{attributedTo.Agent} \tag{2}$$

$$\exists \text{generatedBy.ProvenanceActivity} \sqsubseteq \text{EntityWithProvenance} \tag{3}$$

$$\text{EntityWithProvenance} \sqsubseteq \forall \text{generatedBy.ProvenanceActivity} \tag{4}$$

$$\exists \text{used.EntityWithProvenance} \sqsubseteq \text{ProvenanceActivity} \tag{5}$$

$$\text{ProvenanceActivity} \sqsubseteq \forall \text{used.EntityWithProvenance} \tag{6}$$

$$\exists \text{performedBy.Agent} \sqsubseteq \text{ProvenanceActivity} \tag{7}$$

$$\text{ProvenanceActivity} \sqsubseteq \forall \text{performedBy.Agent} \tag{8}$$

Axiom Explanations

1. Scoped Domain:The scoped domain of attributedTo, scoped by Agent, is EntityWithProvenance.
2. Scoped Range: The scoped range of attributedTo, scoped by EntityWithProvenance, is Agent.
3. Scoped Domain:The scoped domain of generatedBy, scoped by ProvenanceActivity, is EntityWithProvenance.
4. Scoped Range: The scoped range of generatedBy, scoped by EntityWithProvenance, is ProvenanceActivity.
5. Scoped Domain:The scoped domain of used, scoped by EntityWithProvenance, is ProvenanceActivity
6. Scoped Range: The scoped range of used, scoped by ProvenananceActivity, is EntityWithProvenance.
7. Scoped Domain:The scoped domain of performedBy, scoped by Agent, is ProvenanceActivity.
8. Scoped Range: The scoped range of performedBy, scoped by ProvenanceActivity, is Agent.

---

[13] Axiomatization is extensive while avoiding undesirably strong ontological committments. Most axioms for the MODL patterns follow the template of the OWLAx Protégé plug-in [20].

Competency Questions

CQ1. Who are the contributors to this Wikidata page?
CQ2. From which database is this entry taken?
CQ3. Which method was used to generate this chart and from which spreadsheet did the data originate?
CQ4. Who provided this research result?

3.4 Details

Persistent URI The persistent URI for this resource is `https://archive.org/services/purl/purl/modular_ontology_design_library`. The Version 1.0 snapshot and its documentation may be found there. Additionally, it provides helpful links to a technical report and the living data on GitHub, as discussed below.

Canonical Citation The canonical citation for this resource may be found on arXiv [22].

Documentation In addition to this document, we provide in-depth documentation on the library. This documenation contains a primer on ontology design patterns, as a concept, as well as common techniques used in their formalization. Most importantly, for each pattern it provides a schema diagram, its axiomatization, and explanations for each of those axioms. As mentioned in Section 3.1, each pattern is thoroughly annotated with OPLa which provides further documentation on its use and provenance.

Sustainability & Maintenance MODL straddles the realms of dataset and software library; the resource is essentially a snapshot of data that lives. Due to this potential for change, we intend to maintain MODL analogously to a software project. Indeed, while the snapshots will be distributed as ZIP archives, the living data is (at the time of this writing) hosted on GitHub.[14] The Data Semantics Laboratory[15] will host MODL's snapshots and appropriate documentation indefinitely. The authors plan to drive further development of needed or requested patterns. Furthermore, by using Git[16] we inherit mechanisms for tracking issues and versions and incorporating such community contributions into future releases.

License Information This resource is released under the Creative Commons Attribution 4.0 International Public License the details of which can be found online.[17]

---

[14] `https://github.com/cogan-shimizu-wsu/modular-ontology-design-library`
[15] `http://daselab.org/`
[16] `https://git-scm.com/`
[17] `https://creativecommons.org/licenses/by/4.0/legalcode`

4 Conclusions

MODL is a curated collection of well-documented ontology design patterns. We have created this resource to meet a community-recognized need for tooling infrastructure for ontology engineering. In particular, this resource makes ontology design patterns both findable and accessible, shows how they are interoperable, and promotes their reuse. Furthermore, we posit that future ontologies reusing these patterns will promote their interoperability and reuse.

4.1 Next Steps

The next steps are many, as MODL is a foundational resource. We have identified several patterns that we deem necessary for covering additional frequently encountered modelling needs, e.g. a process pattern or patterns. As mentioned in Section 3.1, we also want to further flesh out the documentation with respect to [14]. One future use case that we foresee for this resource is the mapping of competency questions to example SPARQL queries, which maybe could be used as a gold-standard training set for an automated translator. Also mentioned in Section 3.1, we intend to work closely with the digital humanities community for their knowledge representation needs. Finally, we have noted the extreme importance of working closely with tool developers; there is ongoing work with the developer of [8] to create a Protégé plug-in that utilizes MODL as a base for modular ontology modelling.

References

1. Gene ontology consortium: The gene ontology (GO) database and informatics resource. *Nucleic Acids Research*, 32(Database-Issue):258–261, 2004.
2. E. Blomqvist, K. Hammar, and V. Presutti. Engineering ontologies with patterns - the eXtreme Design methodology. In P. Hitzler, A. Gangemi, K. Janowicz, A. Krisnadhi, and V. Presutti, editors, *Ontology Engineering with Ontology Design Patterns - Foundations and Applications*, volume 25 of *Studies on the Semantic Web*, pages 23–50. IOS Press, 2016.
3. P. A. Bonatti, S. Decker, A. Polleres, and V. Presutti. Knowledge Graphs: New Directions for Knowledge Representation on the Semantic Web (Dagstuhl Seminar 18371). *Dagstuhl Reports*, 8(9):29–111, 2019.
4. Z. Dragisic, P. Lambrix, and E. Blomqvist. Integrating ontology debugging and matching into the extreme design methodology. In E. Blomqvist, P. Hitzler, A. Krisnadhi, T. Narock, and M. Solanki, editors, *Proceedings of the 6th Workshop on Ontology and Semantic Web Patterns (WOP 2015) co-located with the 14th International Semantic Web Conference (ISWC 2015), Bethlehem, Pensylvania, USA, October 11, 2015.*, volume 1461 of *CEUR Workshop Proceedings.* CEUR-WS.org, 2015.

5. A. Gangemi, M. Alam, L. Asprino, V. Presutti, and D. R. Recupero. Framester: A wide coverage linguistic linked data hub. In E. Blomqvist, P. Ciancarini, F. Poggi, and F. Vitali, editors, *Knowledge Engineering and Knowledge Management - 20th International Conference, EKAW 2016, Bologna, Italy, November 19-23, 2016, Proceedings*, volume 10024 of *Lecture Notes in Computer Science*, pages 239–254, 2016.

6. A. Gangemi and V. Presutti. Ontology design patterns. In S. Staab and R. Studer, editors, *Handbook on Ontologies*, International Handbooks on Information Systems, pages 221–243. Springer, 2009.

7. T. R. Gruber. A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2):199–220, 1993.

8. K. Hammar. Ontology design patterns in WebProtégé. In S. Villata, J. Z. Pan, and M. Dragoni, editors, *Proceedings of the ISWC 2015 Posters & Demonstrations Track co-located with the 14th International Semantic Web Conference (ISWC-2015), Bethlehem, PA, USA, October 11, 2015.*, volume 1486 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2015.

9. K. Hammar, E. Blomqvist, D. Carral, M. van Erp, A. Fokkens, A. Gangemi, W. R. van Hage, P. Hitzler, K. Janowicz, N. Karima, A. Krisnadhi, T. Narock, R. Segers, M. Solanki, and V. Svátek. Collected research questions concerning ontology design patterns. In P. Hitzler, A. Gangemi, K. Janowicz, A. Krisnadhi, and V. Presutti, editors, *Ontology Engineering with Ontology Design Patterns - Foundations and Applications*, volume 25 of *Studies on the Semantic Web*, pages 189–198. IOS Press, 2016.

10. P. Hitzler, A. Gangemi, K. Janowicz, A. Krisnadhi, and V. Presutti, editors. *Ontology Engineering with Ontology Design Patterns – Foundations and Applications*, volume 25 of *Studies on the Semantic Web*. IOS Press, 2016.

11. P. Hitzler, A. Gangemi, K. Janowicz, A. A. Krisnadhi, and V. Presutti. Towards a simple but useful ontology design pattern representation language. In E. Blomqvist, Ó. Corcho, M. Horridge, D. Carral, and R. Hoekstra, editors, *Proceedings of the 8th Workshop on Ontology Design and Patterns (WOP 2017) co-located with the 16th International Semantic Web Conference (ISWC 2017), Vienna, Austria, October 21, 2017.*, volume 2043 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2017.

12. P. Hitzler and A. Krisnadhi. A tutorial on modular ontology modeling with ontology design patterns: The cooking recipes ontology. *CoRR*, abs/1808.08433, 2018.

13. N. Karima. *Automated Rendering of Schema Diagram for Ontologies*. PhD thesis, Wright State University, 2017.

14. N. Karima, K. Hammar, and P. Hitzler. How to document ontology design patterns. In K. Hammar, P. Hitzler, A. Lawrynowicz, A. Krisnadhi, A. Nuzzolese, and M. Solanki, editors, *Advances in Ontology Design and Patterns*, volume 32 of *Studies on the Semantic Web*, pages 15–28. IOS Press, Amsterdam, 2017.

15. A. Krisnadhi and P. Hitzler. Modeling with ontology design patterns: Chess games as a worked example. In P. Hitzler, A. Gangemi, K. Janowicz, A. Krisnadhi, and V. Presutti, editors, *Ontology Engineering with Ontology Design Patterns – Foundations and Applications*, volume 25 of *Studies on the Semantic Web*, pages 3–21. IOS Press, 2016.

16. A. Krisnadhi, Y. Hu, K. Janowicz, P. Hitzler, R. A. Arko, S. Carbotte, C. Chandler, M. Cheatham, D. Fils, T. W. Finin, P. Ji, M. B. Jones, N. Karima, K. A. Lehnert, A. Mickle, T. W. Narock, M. O'Brien, L. Raymond, A. Shepherd, M. Schildhauer, and P. Wiebe. The geolink modular oceanography ontology. In M. Arenas, Ó. Corcho, E. Simperl, M. Strohmaier, M. d'Aquin, K. Srinivas, P. T. Groth, M. Dumontier, J. Heflin, K. Thirunarayan, and S. Staab, editors, *The Semantic Web - ISWC*

2015 - 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part II, volume 9367 of *Lecture Notes in Computer Science*, pages 301–309. Springer, 2015.

17. V. Presutti, E. Daga, A. Gangemi, and E. Blomqvist. eXtreme Design with content ontology design patterns. In E. Blomqvist, K. Sandkuhl, F. Scharffe, and V. Svátek, editors, *Proceedings of the Workshop on Ontology Patterns (WOP 2009) , collocated with the 8th International Semantic Web Conference (ISWC-2009), Washington D.C., USA, 25 October, 2009.*, volume 516 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2009.

18. A. L. Rector. Modularisation of domain ontologies implemented in description logics and related formalisms including OWL. In J. H. Gennari, B. W. Porter, and Y. Gil, editors, *Proceedings of the 2nd International Conference on Knowledge Capture (K-CAP 2003), October 23-25, 2003, Sanibel Island, FL, USA*, pages 121–128. ACM, 2003.

19. S. Sahoo, D. McGuinness, and T. Lebo. PROV-o: The PROV ontology. W3C recommendation, W3C, Apr. 2013. http://www.w3.org/TR/2013/REC-prov-o-20130430/.

20. M. K. Sarker, A. A. Krisnadhi, and P. Hitzler. OWLAx: A protégé plugin to support ontology axiomatization through diagramming. In T. Kawamura and H. Paulheim, editors, *Proceedings of the ISWC 2016 Posters & Demonstrations Track co-located with 15th International Semantic Web Conference (ISWC 2016), Kobe, Japan, October 19, 2016.*, volume 1690 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2016.

21. C. Shimizu, A. Eberhart, N. Karima, Q. Hirt, A. Krisnadi, and P. Hitzler. A method for automatically generating schema diagrams for modular ontologies. In *1st Iberoamerican Conference on Knowledge Graphs and the Semantic Web*, 2019. To Appear.

22. C. Shimizu, Q. Hirt, and P. Hitzler. Modl: A modular ontology design library. Technical report, Wright State University, Dayton, Ohio, April 2019.

23. C. Shimizu, P. Hitzler, Q. Hirt, A. Shiell, S. Gonzalez, C. Foley, D. Rehberger, E. Watrall, W. Hawthorne, D. Tarr, R. Carty, and J. Mixter. The enslaved ontology 1.0: People of the historic slave trade. Technical report, Michigan State University, East Lansing, Michigan, April 2019.

24. C. Shimizu, P. Hitzler, and C. Paul. Ontology design patterns for winston's taxonomy of part-whole relations. In E. Demidova, A. Zaveri, and E. Simperl, editors, *Emerging Topics in Semantic Technologies – ISWC 2018 Satellite Events [best papers from 13 of the workshops co-located with the ISWC 2018 conference]*, volume 36 of *Studies on the Semantic Web*, pages 119–129. IOS Press, 2018.

25. M. G. Skjæveland, D. P. Lupp, L. H. Karlsen, and H. Forssell. Practical ontology pattern instantiation, discovery, and maintenance with reasonable ontology templates. In D. Vrandecic, K. Bontcheva, M. C. Suárez-Figueroa, V. Presutti, I. Celino, M. Sabou, L. Kaffee, and E. Simperl, editors, *The Semantic Web - ISWC 2018 - 17th International Semantic Web Conference, Monterey, CA, USA, October 8-12, 2018, Proceedings, Part I*, volume 11136 of *Lecture Notes in Computer Science*, pages 477–494. Springer, 2018.

26. M. D. Wilkinson, M. Dumontier, et al. The fair guiding principles for scientific data management and stewardship. *Scientific Data*, 3:160018 EP –, Mar 2016. Comment.