Considerations regarding Ontology Design Patterns

Editorial

Eva Blomqvist [a], Pascal Hitzler [b], Krzysztof Janowicz [c], Adila Krisnadhi [b,d], Tom Narock [e], Monika Solanki [f]

[a] *Linköping University, Sweden*
[b] *Wright State University, Dayton, OH, USA*
[c] *University of California, Santa Barbara, USA*
[d] *Faculty of Computer Science, Universitas Indonesia*
[e] *Marymount University, USA*
[f] *University of Oxford, UK*

1. Introduction

The Worskhop on Ontology and Semantic Web Patterns (WOP2015, 6th edition)[1] was held on October 11, 2015 in conjunction with the 14th International Semantic Web Conference in Bethlehem, PA, USA. At the workshop, the organizers conducted a discussion with the participants regarding the promises and obstacles of ontology design patterns (ODPs for short). This editorial reports on those discussions. We begin with a brief introduction of ODPs for the unfamiliar reader.

2. Ontology Design Patterns: A Brief Primer

An ontology design pattern, generally speaking, is a "reusable successful solution to a recurrent [ontology] modeling problem" [17], such as how to represent ternary predicates in OWL and RDF, or how to model generic notions such as *event* or *process*. The former are known as *logical patterns*, while the latter are known as *content patterns*. Several other classes of patterns have also been suggested [18].

Content patterns can be understood as ontology snippets, or parts, which capture key and widely reusable aspects of a single notion such as *event*, *organization*, or *trajectory* [11]. A well-designed content pattern should be very versatile, i.e. reusable, in many different application contexts. This requirement is often achieved by avoiding overly strong (i.e., application-specific or potentially controversial) ontological commitments.

The different roles agents can play (such as as in an organization or event) is an example of a content pattern. A person (as agent) may be member of an organization or may be author of a paper. An organization, like IOS Press, may have the role of publisher for a journal such as the Semantic Web journal. The corresponding pattern is sometimes referred to as the *AgentRole* pattern.

Figure 1 depicts the structure of this pattern as used in the GeoLink Modular Ontology (GMO) [15]. The classes TimeInstant and Agent represent complex types of entities in their own right that would typically be modeled by corresponding additional patterns. These additional patterns can then be connected with the AgentRole pattern as part of an ontology construction process.

Patterns are usually formally expressed through a logical axiomatization, such as the Web Ontology Language OWL [9], which defines the formal semantics and relationships between the vocabulary items used in the pattern. Patterns may also come with a set of mappings that explain the for-

[1] http://ontologydesignpatterns.org/wiki/WOP:2015

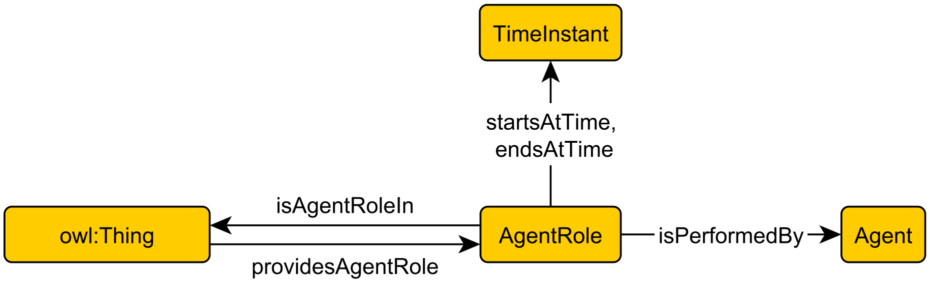

Fig. 1. Depiction of the AgentRole pattern.

mal relationship between the pattern and other established patterns, ontologies, or controlled vocabularies, some of which can also be expressed using OWL or rules.

The GMO axiomatization of this pattern[2] includes the axioms (in description logic syntax [10]) listed in Figure 2. Axiom (1) states that an AgentRole is performed by exactly one Agent, has exactly one starting time and one ending time, and is an agent role in exactly one thing. Axiom (2) states that providesAgentRole and isAgentRoleIn are inverse properties. Other axioms of the pattern give, for example, (guarded) domain and range restrictions for the properties of the pattern.

Ontology design patterns were introduced independently by Blomqvist and Gangemi [3,4] in 2005. Ontology engineering with ODPs was described in the eXtreme Design methodology [19] as well as ODOE [12] and has been systematically practiced at the U.S. GeoVoCamps since 2012 [8]. The above mentioned GeoLink Modular Ontology [15] is an example for a recent ontology constructed using ODP modeling principles.

3. Promises of Ontology Design Patterns

3.1. ODPs as an Ontology Engineering Tool

Arguments for the benefits of ODPs focus mainly on the ontology engineering process and, indeed, using libraries of design patterns has proven beneficial in areas such as software engineering. Patterns act as a language for talking about problems and solutions; collecting benefits, drawbacks, and consequences; and providing examples of design classes. All the while building on the old principle of not having to reinvent the wheel.

At first glance, novice ontology engineers may feel that ontology engineering is as easy as drawing a diagram on paper. However, it is soon discovered that logical modeling, e.g., using OWL, can be as complex as programming, due to the underlying semantics of the language. This warrants the use of simple ODPs that can help to avoid common pitfalls or simply assist in making informed design choices [1,2]. Although, this original motivation for ODPs is still not completely realized in practice. Software Engineering, by comparison, has a much more coherent and consensual set of patterns available, carefully described in books and online catalogues, and taught at universities. This is not yet the case for ODPs. However, the process is certainly ongoing. As ODPs become more mature and the community collects more and more experience using them, we envision that the same situation as in Software Engineering will emerge in Ontology Engineering – ODPs as one of the cornerstones of Ontology Engineering.

Similar to Software Engineering, ODPs also have additional benefits going beyond being abstract descriptions of common problem-solution pairs. Their difference, however, originates in the difference between programming and modeling. While software design patterns are quite abstract, sometimes a long way from their actual implementation in a certain programming language, ODPs may in many cases be quite close in level of abstraction and representation to their implementation in an ontology modeling language such as OWL. This means it is often possible to describe the ODP itself as a reusable component using

[2] See http://schema.geolink.org/patterns/

$$\begin{aligned}\mathsf{AgentRole} \sqsubseteq\ & (=1\ \mathsf{isPerformedBy.Agent}) \sqcap (=1\ \mathsf{isAgentRoleIn}.\top) \\ & \sqcap (=1\ \mathsf{timeStarts.TimeInstant}) \sqcap (=1\ \mathsf{timeEnds.TimeInstant}) \end{aligned} \tag{1}$$

$$\mathsf{providesAgentRole} \equiv \mathsf{isAgentRoleIn}^{-} \tag{2}$$

Fig. 2. Some axioms for the AgentRole pattern.

the implementation language (e.g., OWL) directly. This facilitates the use of ODPs not only as abstract ideas, but also as actual reusable components. Ontology Engineering could then be seen more as a composition task, i.e., composing a set of reusable ODP implementations, rather than building ontologies from scratch. This of course sets requirements on the quality and documentation of the ODP components. The ontology engineer needs to be aware of the consequences of reusing such components and how to make correct connections between the components being integrated. ODPs bring a promise of compositional modeling and true ontology reuse, which has the potential of considerably reducing time and effort spent in some Ontology Engineering tasks.

3.2. ODPs for Improved Interoperability

One of the original visions of ontology reuse was that shared Web ontologies would enable large-scale data integration. Web-wide data integration would be possible from multiple datasets reusing the ontologies as schemas. However, in reality, most datasets on the Web, e.g. Linked Data, use their own ontology, or reuse (and potentially misuse) a mismatch of concepts and properties from various ontologies and less expressive vocabularies. In practice, reusing a complete ontology is often infeasible, due to some (potentially small) part of it not matching your data or conceptualization. For example, most ontologies make too strong ontological commitments to be directly reused. ODPs seem promising as a middle ground between reusing a complete ontology and making your own model from scratch. ODPs are more likely to fit your data and conceptualization of the world than large upper ontologies since ODPs can be viewed as small components (i.e., small ontologies) with minimal ontological commitments. By reusing an ODP you get a certain level of interoperability of your data with others reusing the same ODP. This interoperability may not address every detail, but the ODP ensures a minimal level of interoperability where data can actually be integrated Web-wide. This is a promise of real Linked Data integration, which is at the moment desperately needed [13,21].

Turning that argument around, this also means that heterogeneity is still allowed. ODPs support heterogeneity as they only ensure, and enforce, a minimal level of interoperability. This means that we can still cater to specifics within each dataset that are bound to occur. Thus, ODPs enhance reusability while simultaneously facilitating practical use of in real-world data and applications [15].

Using ODPs as the basis for both ontologies, and less complex Linked Data vocabularies, will also lead to better understanding of ontologies and datasets as ODPs can be seen as part of the documentation of the ontology or dataset. Making explicit that you use an ODP, or even reusing its component implementation by importing it, sends a clear message to anyone that later studies the ontology or dataset. The ODP, as a common language for ontology engineers, along with its documentation can give a better understanding of the underlying conceptualization and ontological commitments. This can create a natural interoperability on the ontology level. By identifying that two ontologies use the same ODP they have a natural, and inherent, point of alignment.

3.3. ODPs for Improved Application Support

A common way of publishing Linked Data on the Web is to create a vocabulary completely bottom-up. Publishers often look at the existing data structure and simply replicate it using concepts and relations that are found in existing vocabularies (or created in a new custom vocabulary). This straight forward method ensures that data gets out on the Web. Yet, it is not necessarily a method that creates good datasets from a reuse and application perspective. This is another area where ODPs promise improvement [21].

ODPs are, although generic in some sense, usually developed together with domain experts. For instance, ODPs for geographical data are usually developed together with geographers, Geographic Information System (GIS) specialists, and other domain experts. This ensures that application requirements of common applications in the domain get built into the ODPs. In turn, data publishers using those ODPs for their ontologies are aware of those requirements and if possible cater for them in their data.

Applications themselves (or application developers) could also use ODPs to understand the potential of data, i.e., what the data contains and what can be done with it. This is certainly true for the use of ontologies as schemas in general. Yet, having a limited set of ODPs to cater to in an application is most likely more feasible than building an application that can adapt to all the opportunities of the (potentially unlimited) set of arbitrarily modeled ontologies.

3.4. ODP Support

ODPs bring to Ontology Engineering the promise of data and ontology publishing and reuse. Yet, this can not be realized without work on existing challenges within the ODP community. Finding ways to assess and ensure ODP quality is one such area that is currently being actively researched (e.g., as in [6]). Having clear criteria for the quality of an ODP, methods to assess quality, and methods to select ODPs based on different quality criteria, promises to greatly improve the reliability of ODP catalogues and the ontologies that later use them. Similarly, methods and tools for more easily finding and reusing ODPs (e.g., as discussed in [7]), including understanding the relations between ODPs, such as identifying alternative and complementing ODPs, promises to greatly improve the usability of ODPs in Ontology Engineering. Conventions for making ODP use explicit, even when ODP implementation components are not directly imported into the ontology, can further increase interoperability and application support.

4. Barriers to the widespread adoption of Ontology Design Patterns

The promises offered by ODPs provide a reasonably sound justification for their usage. Yet, many barriers to their adoption still remain to be overcome before ODPs become a part of mainstream ontology engineering.

Availability of relevant patterns. One of the most widespread criticisms against the use of ODPs is the lack of relevant patterns, both at a generic and a domain specific level. While two public ontology libraries: ODP-Wiki and Manchester ODPs Public Catalog for bio-ontologies (MBOP), have been made available, each describing the purpose of the pattern and its formalization, finding patterns that have been validated against specific requirements or use cases has still remained a largely hard task.

Pattern discovery services. The capability to uniformly look up ODPs on demand, using a service that provides machine processable pattern metadata and APIs, has long since been a desideratum for ontology authors and editors. This too has often been cited as one of the major barriers to the uptake of ODPs. A standardized mechanism for semantically describing ODPs that can be exploited by ontology discovery engines and facilitate their automated discovery and usage by agents needs to be made available.

Minimum tooling support. Existing ontology editing environments provide the mechanism to import and instantiate ontologies. Yet, no explicit support is currently available to develop and publish patterns directly to public ODP servers. Additionally, there is no search support for patterns in public ODP libraries. The problem becomes significantly complex when domain experts, relying heavily on tool support, need solutions to recurring problems in specific domains. The lack of tools that facilitate and interactively prompt the use of patterns while an ontology is being authored is a key bottleneck. Further, editors that proactively analyze an ontology while importing it and recognize the use of well documented patterns are much needed.

Legacy ontologies, hidden patterns. Many well known and widely used ontologies were developed before ODPs were introduced. Some of these ontologies are upper level such as BFO[3] and SUMO[4], while others such as SNOMED-CT[5] and CIDOC-CRM[6] are relatively large. Such ontologies potentially encode a number of ODPs. A fact which is implicitly validated by their widespread usage. However, analyzing and restructuring these ontologies to identify patterns is a non-trivial task requiring significant support, which is often not available. Consequently, many hidden ODPs, which could be extracted and utilized to support ontology engineering, thereby bridging the gap between the demand and availability of ODPs, remain out of reach.

Strategies for pattern design development. The development of Software Engineering design patterns has conventionally followed a bottom-up approach where recurring patterns are extracted from existing software. This also serves as a validation of the usage of the pattern. However, the development of ODPs has largely happened in a top-down manner. This was justified in the early days of ontology development due to a lack of critical mass. Yet, strategies are now required that encourage bottom-up development and enable automated extraction of patterns from a corpus of ontologies.

Pattern evaluation and validation. ODP development suffers from a lack of benchmarks and quality heuristics and metrics against which patterns can be evaluated. This is a critical requirement in order to provide increased confidence in their usage. Thus, while ODPs provide a compositional approach to ontology design, validation of the composition has so far been largely ignored.

Expressivity conflicts. Conflicts arise in several scenarios due to the high levels of expressivity provided by certain patterns and the need for minimum ontological commitment required by ontologies. Patterns are intended to aid the designer, especially in the design of large ontologies and the reduction in scope of errors. If the inclusion of ODPs alters the desired expressivity, making the overall ontology intractable, there is bound to be a reluctance to their adoption.

5. Possible Research Directions

Participants at the workshop were also asked to share their thoughts on key research directions that should be pursued. In the following, we list what emerged as main thrusts during the short time we had for this part of the discussion. This collection is by no means exhaustive. Additional current research questions have already been alluded to in previous sections of this paper.

Investigate causes for poorly designed ontologies. Most ontologies are rarely, if ever, reused by third parties. This flies in the face of one of the original main motivations for creating ontologies, namely as shared conceptualizations. However, the exact reasons why ontologies find so little reuse are at this point very poorly understood. Anecdotal evidence indicates that ontologies often are poorly designed and constructed, difficult to understand, insufficiently documented and maintained, too specific or too broad (or both). Social dynamics and findability may also play a role.

As ontology design patterns are created and used, it behooves the community to really understand the causes for the lack of reuse. In particular, the ODP community should be able to address possible causes related to poor design, to understandability and documentation issues, and to finding an appropriate balance for specificity versus generality. A thorough and evidence-supported understanding of these issues, e.g. what exactly “poor design” and “good design” are, appears to be necessary to advance on these fronts; investigations into ontological anti-patterns fall into this realm [22]. In particular, there seems to be a lack of user-centric evaluations addressing the ontology reuse issue, with only a few notable exceptions, e.g. [14].

Produce a critical amount of reusable ontology design patterns. The advance of ontology design patterns is caught in a catch-22: In order to provide convincing evidence for the added value of ODPs, the community requires access to a well-organized, well-documented, and well-maintained set of high-quality ontology design patterns. At the same time, however, there is a lack of incen-

[3] http://ifomis.uni-saarland.de/bfo/

[4] http://www.adampease.org/OP/

[5] http://bioportal.bioontology.org/ontologies/SNOMEDCT

[6] http://www.cidoc-crm.org/rdfs/cidoc_crm_v5.0.4_official_release.rdfs

tive (and funding) to provide these, as long as this added value has not yet been convincingly demonstrated.

A joint effort to create, document, and properly catalogue key ontology design patterns will be needed, as well as a discussion on which patterns are needed and how they should be provided. An pattern creation effort is ongoing for the geosciences domain, in form of the U.S. GeoVocamps [8], but the process is still very ad-hoc, and a broader organized effort will be needed.

Develop an ontology design pattern representation language. Currently, ontology design patterns are mostly presented in the form of OWL files, with some minimalistic accompanying documentation. While OWL can indeed be used to express the core of a pattern, it does not provide native ways to represent additional information which would be helpful for reuse or organization of patterns. For example, it would seem to be important to record how different patterns relate to each other on a more abstract level, e.g. whether a pattern is a refinement or a generalization of another pattern. Using OWL to express axiomatizations provided with patterns is also limiting because the open-world assumption underlying OWL does not cater for expressing non-monotonic constructs like constraints, and because some patterns call for an axiomatization of, say, transitive closure or general rules not expressible in OWL DL. Some first steps towards developing such a representation language have already been undertaken [5] but more work remains to be done.

Formalize relationships between patterns. Related to the previous point is the question of understanding, in depth, how different patterns can relate to each other. This is important to understand compositionality aspects, e.g. how to create an ontology based on existing ontology design patterns. Key relations are specialization, generalization, and composition and are discussed as in [20], but often relationships are more complex. For example, there are notions of *views* or *shortcuts*, see [8], which can be understood as a type of temporary simplification of a pattern.

Understanding relevant notions of how patterns relate to each other seems to be required in order to develop a practically useful pattern representation language, and also in order to develop methodologies and tools which would be able to lend significant support to ontology engineers.

Sustainability issues related to ontology design patterns. The provision, maintenance and documentation of ontology design patterns are currently done in a very ad-hoc manner. While the community has a rudimentary portal,[7] it lends only limited support and structure. Versioning aspects and other software engineering related issues need to be addressed, and better tool support for pattern-driven ontology engineering needs to be developed.

Thus, sustainability related research questions are about finding how to best implement infrastructure and apply software engineering principles to ODPs. What would a tool for the efficient support of ODP-driven ontology-engineering look like? How should documentation be provided? What does versioning mean for ontologies or ODPs on the Web?

Ontology design patterns for data publishing and reuse, and other use cases. Good, and highly visible, use cases can be powerful drivers of community efforts. While such use cases for pattern-driven ontology engineering exist (e.g. the recent [15,16]), they need to be catalogued, made accessible, and assessed regarding the relevance of ODPs to the scenario. Potential additional use cases could be developed with focus on data publishing and reuse, e.g. in the realm of linked data [21]. Perhaps some type of regular *challenge* at the WOP workshops could be set up to facilitate this and other research aspects mentioned above.

6. Conclusions

WOP2015 served as a gathering point for the ontology design pattern community. It provided an arena for proposing and discussing best practices, patterns, pattern-based ontologies, and pattern-based systems. Workshop posters and presentations highlighted many of the aforementioned benefits of ODPs and several new patterns were introduced to the community. Many successes were presented and discussed. Yet, as noted here, challenges remain to full fledged adoption of the ODP methodology. It is our hope that this editorial summaries the ODP benefits and challenges for

[7] http://www.ontologydesignpatterns.org

the broader Semantic Web community. We look forward to the next steps in research and development of ODPs following WOP2015.

Acknowledgements. Hitzler, Janowicz, Krisnadhi, and Narock acknowledge funding by the National Science Foundation under award 1440202 *EarthCube Building Blocks: Collaborative Proposal: GeoLink – Leveraging Semantics and Linked Data for Data Sharing and Discovery in the Geosciences.* The authors furthermore thank the 20 participants of the discussion at WOP2015, which included Yingjie Hu (UCSB), Da Huo, Holly Ferguson, Charles Vardeman (University of Notre Dame), Christophe Debruyne (Trinity College Dublin), Agnieszka Lawrynowicz (Poznan University of Technology), Nick Raphael (Harvard Business School Baker Library, David Carral (Wright State University), Quang-Khai Tran (Korea Institute of Science and Technology Information) and Takahiro Kawamura.

References

[1] E. Blomqvist, A. Gangemi, and V. Presutti. Experiments on pattern-based ontology design. In *Proceedings of the 5th International Conference on Knowledge Capture (K-CAP 2009), September 1-4, 2009, Redondo Beach, California, USA*, pages 41–48. ACM, 2009. doi:10.1145/1597735.1597743.

[2] E. Blomqvist, V. Presutti, E. Daga, and A. Gangemi. Experimenting with eXtreme Design. In P. Cimiano and H. S. Pinto, editors, *Knowledge Engineering and Management by the Masses – 17th International Conference, EKAW 2010, Lisbon, Portugal, October 11-15, 2010. Proceedings*, volume 6317 of *Lecture Notes in Computer Science*, pages 120–134. Springer, 2010. doi:10.1007/978-3-642-16438-5_9.

[3] E. Blomqvist and K. Sandkuhl. Patterns in ontology engineering: Classification of ontology patterns. In *ICEIS 2005, Proceedings of the Seventh International Conference on Enterprise Information Systems, Miami, USA, May 25-28, 2005*, pages 413–416, 2005.

[4] A. Gangemi. Ontology design patterns for semantic web content. In Y. Gil, E. Motta, V. R. Benjamins, and M. A. Musen, editors, *The Semantic Web – ISWC 2005, 4th International Semantic Web Conference, ISWC 2005, Galway, Ireland, November 6-10, 2005, Proceedings*, volume 3729 of *Lecture Notes in Computer Science*, pages 262–276. Springer, 2005. doi:10.1007/11574620_21.

[5] G. Guizzardi. Ontological patterns, anti-patterns and pattern languages for next-generation conceptual modeling. In E. S. K. Yu, G. Dobbie, M. Jarke, and S. Purao, editors, *Conceptual Modeling – 33rd International Conference, ER 2014, Atlanta, GA, USA, October 27-29, 2014. Proceedings*, volume 8824 of *Lecture Notes in Computer Science*, pages 13–27. Springer, 2014. doi:10.1007/978-3-319-12206-9_2.

[6] K. Hammar. *Towards an Ontology Design Pattern Quality Model.* Linköping Studies in Science and Technology. Linköping University Electronic Press, 2013.

[7] K. Hammar. Ontology design patterns: Improving findability and composition. In *The Semantic Web: ESWC 2014 Satellite Events – ESWC 2014 Satellite Events, Anissaras, Crete, Greece, May 25-29, 2014, Revised Selected Papers*, volume 8798 of *Lecture Notes in Computer Science*, pages 3–13. Springer, 2014. doi:10.1007/978-3-319-11955-7_1.

[8] P. Hitzler, K. Janowicz, and A. Krisnadhi. Ontology modeling with domain experts: The GeoVoCamp experience. In C. d'Amato, F. Lecue, R. Mutharaju, T. Narock, and F. Wirth, editors, *Proceedings of the Diversity++ Workshop co-located with the 14th International Semantic Web Conference, ISWC 2015 October 12th, 2015, Bethlehem, Pennsylvania, USA*, 2015. To appear.

[9] P. Hitzler, M. Krötzsch, B. Parsia, P. F. Patel-Schneider, and S. Rudolph, editors. *OWL 2 Web Ontology Language: Primer (Second Edition).* W3C Recommendation 11 December 2012, 2012. Available at http://www.w3.org/TR/owl2-primer.

[10] P. Hitzler, M. Krötzsch, and S. Rudolph. *Foundations of Semantic Web Technologies.* Chapman & Hall/CRC, 2010.

[11] Y. Hu, K. Janowicz, D. Carral, S. Scheider, W. Kuhn, G. Berg-Cross, P. Hitzler, and M. Dean. A geo-ontology design pattern for semantic trajectories. In T. Tenbrink, J. G. Stell, A. Galton, and Z. Wood, editors, *Spatial Information Theory – 11th International Conference, COSIT 2013, Scarborough, UK, September 2-6, 2013. Proceedings*, volume 8116 of *Lecture Notes in Computer Science*, pages 438–456, Heidelberg, 2013. Springer. doi:10.1007/978-3-319-01790-7_24.

[12] K. Janowicz. Observation-driven geo-ontology engineering. *Transactions in GIS*, 16(3):351–374, 2012.

[13] K. Janowicz, F. van Harmelen, J. A. Hendler, and P. Hitzler. Why the data train needs semantic rails. *AI Magazine*, 26(1):5–14, 2015.

[14] C. M. Keet. The use of foundational ontologies in ontology development: An empirical assessment. In G. Antoniou, M. Grobelnik, E. P. B. Simperl, B. Parsia, D. Plexousakis, P. D. Leenheer, and J. Z. Pan, editors, *The Semantic Web: Research and Applications – 8th Extended Semantic Web Conference, ESWC 2011, Heraklion, Crete, Greece, May 29-June 2, 2011, Proceedings, Part I*, volume 6643 of *Lecture Notes in Computer Science*, pages 321–335. Springer, 2011. doi:10.1007/978-3-642-21034-1_22.

[15] 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. 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. doi:10.1007/978-3-319-25010-6_19.

[16] A. A. Krisnadhi, Y. Hu, K. Janowicz, P. Hitzler, R. Arko, S. Carbotte, C. Chandler, M. Cheatham, D. Fils, T. Finin, P. Ji, M. Jones, N. Karima, K. Lehnert, A. Mickle, T. Narock, M. O'Brien, L. Raymond, A. Shepherd, M. Schildhauer, and P. Wiebe. The GeoLink framework for pattern-based linked data integration. In *Proceedings of the Posters and Demos session at the 14th International Semantic Web Conference (ISWC 2015), Bethlehem, Pensylvania, USA, October 2015*, 2015.

[17] Ontology Design Patterns . org (ODP). http://ontologydesignpatterns.org, May 2015. Online; accessed on October 28, 2015.

[18] Ontology design patterns types. http://ontologydesignpatterns.org/wiki/OPTypes, May 2010. Online; accessed on October 28, 2015.

[19] V. Presutti, E. Daga, A. Gangemi, and E. Blomqvist. eXtreme Design with Content Ontology Design Patterns. In *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.

[20] V. Presutti and A. Gangemi. Content ontology design patterns as practical building blocks for web ontologies. In Q. Li, S. Spaccapietra, E. S. K. Yu, and A. Olivé, editors, *Conceptual Modeling – ER 2008, 27th International Conference on Conceptual Modeling, Barcelona, Spain, October 20-24, 2008. Proceedings*, volume 5231 of *Lecture Notes in Computer Science*, pages 128–141. Springer, 2008. doi:10.1007/978-3-540-87877-3_11.

[21] V. Rodriguez-Doncel, A. A. Krisnadhi, P. Hitzler, M. Cheatham, N. Karima, and R. Amini. Pattern-based Linked Data publication: The Linked Chess Dataset case. In O. Hartig, J. Sequeda, and A. Hogan, editors, *Proceedings of the 6th International Workshop on Consuming Linked Data co-located with 14th International Semantic Web Conference (ISWC 2105), Bethlehem, Pennsylvania, US, October 12th, 2015*, volume 1426 of *CEUR Workshop Proceedings*, 2015.

[22] T. P. Sales and G. Guizzardi. Ontological anti-patterns: empirically uncovered error-prone structures in ontology-driven conceptual models. *Data and Knowledge Engineering*, 99:72–104, 2015. doi:10.1016/j.datak.2015.06.004.