Pattern-Based Linked Data Publication: The Linked Chess Dataset Case

Víctor Rodríguez-Doncel[1], Adila A. Krisnadhi[2,3], Pascal Hitzler[2], Michelle Cheatham[2], Nazifa Karima[2], and Reihaneh Amini[2]

[1] Ontology Engineering Group, Universidad Politécnica de Madrid
[2] Data Semantics Lab, Wright State University
[3] Faculty of Computer Science, Universitas Indonesia

Abstract. This paper discusses the relationship between ontology design patterns (ODPs), data models and linked data, proposing a method that simplifies the task of publishing linked data while adhering to good modeling practices that reuse well-studied ODPs. The method consists of providing the pattern along with a simplified view of it, the conversion queries to transform one into the other and the constraints to verify pattern conformance. The proposed process simplifies the tasks of the domain experts but preserves the integrity of the design patterns, favoring a well-designed and well documented data model which fosters data reuse. The work is illustrated with a linked dataset of chess games with the key information mapped to other linked datasets and supported by formalized design patterns. A chess dataset is presented as linked data.

Keywords: Ontology Design Patterns, Linked Data, Chess

1 Introduction

In the linked data publication paradigm, datasets typically consist of RDF statements connecting resources from different datasets. These statements may describe the resources using properties and classes further specified in ontologies, thus grounding the data model with a formal specification.

However, linked data publishers are very often not experts in ontology modeling and reuse, e.g. they may lack the time or resources for grounding the vocabularies in use with well-designed formal models, they assume unintended implications (for example abusing of `owl:sameAs` [6]) or their imprecise semantics lead to errors in specific domains [8]. Also, as a result of publishing the data as a simple transformation of existing relational databases or spreadsheets they often miss the benefits of making explicit the details of their data models [7], in particular, easier reuse of the datasets and easier integration of their datasets with others e.g. for federated query answering [10]. The problem of simultaneously satisfying ontology engineers and data publishers is a manifestation of a well studied problem: the choice of the adequate *ontological commitment*. The degree of ontological commitment determines which entities or kinds of entities can and must exist according to a given theory or discourse, and was first studied as a

```
PREFIX sh: <http://www.w3.org/ns/shacl#>
PREFIX sem:   <http://semanticweb.cs.vu.nl/2009/11/sem/>
PREFIX chess-o:   <http://purl.org/NET/rdfchess/ontology/>
PREFIX chess: <http://chessdata.org/rdfchess/resource/>
PREFIX game: <http://chessdata.org/rdfchess/resource/chessgame/>
PREFIX location: <http://chessdata.org/rdfchess/resource/location/>
PREFIX player: <http://chessdata.org/rdfchess/resource/chessplayer/>
PREFIX opening: <http://chessdata.org/rdfchess/resource/opening/>
```

Listing 1. Prefixes used other than RDF, RDFS, SKOS

philosophical problem [13] and then revisited by computer ontologists [4,5] and linked data publishers [2].

Ontology design patterns (*ODPs*) were proposed independently by Blomqvist [1] and Gangemi [3] with the aim of easing the task of designing ontologies by reusing well-studied solutions (or *patterns*) to solve problems which appear recurrently. Some of the ontology design patterns are considered *logical* (structural) patterns and some *content* patterns. The common representation of n-ary predicates by way of several binary predicates is a typical example of logical pattern[4], as well as the idea of using *roles* (in the sense of schema.org[5]) in order to provide a uniform representation for a group of related relationships. Content patterns are patterns that capture an appropriate graph structure for a central notion, such as *person* or *organization*. Understood in a very straightforward way, the abundant use of `foaf:person` or `foaf:name` in the Web of Data can be understood as constituting a kind of pattern, as well as the use of SKOS constructs. These small patterns may lack formal semantics, as the RDF `rdf:List` construct does, but the reuse of familiar structures is useful for understanding, querying and reusing datasets. However, content design patterns more typically provide a well founded axiomatization with a high degree of ontological commitment.

Five-star linked data can be created without giving the exact graph structure much thought, but if the graph structure comes with an underlying logical axiomatization which disambiguates meaning, then consistent reuse (the ultimate goal of linked data) is even more simple. We are neither advocating (1) the adoption of large ontologies as schema for publishing linked datasets, nor are we arguing for (2) standardizing many notions by means of ontologies. Regarding (1) we in fact argue that the focus should be on small, easily understandable and flexible content ontology design patterns, and that of course reuse does not prevent modifying a pattern: if only some recognizable part of the original pattern remains, then this is already of additional benefit. Regarding (2) we argue that use and reuse on the Web should play it out: Useful patterns and graph structures will find reuse probably establishing de-facto standards sooner or later [9].

Adopting content ODP does not necessarily imply that complex data structures are needed to publish linked data: we propose the concept of *view* as a simplified data model keeping the essence of the pattern. And we hold that the view as a simplified ODP is not a final closed model that cannot grow, but which

[4] `http://ontologydesignpatterns.org/wiki/Submissions:N-Ary_Relation_Pattern_%28OWL_2%29`
[5] `https://schema.org/Role`

can be expanded on demand. The central thesis of this paper is that *data models in linked data can be supported by ontology design patterns at low cost by defining simple views and the methods to transform from full pattern-based data structures to simple view and viceversa.*

The description of these ontology views is given in Section 2.1, and the two operations of *Pattern Flattening* and *View Expansion* is given in Section 2.2. The advantages of this approach are listed in Section 3 and illustrated with the Linked Chess Dataset in Section 4.

2 Pattern-Based Linked Data

2.1 Ontology Design Patterns and Ontology Views

As a simple example, one might think of representing a chess game in OWL as an instance of a certain `chess-o:ChessGame` class, which is attributed the literal "Bobby Fischer" as the name of the white player, resulting in Dataset A given in Listing 2. This paper assumes that in many cases, linked data publishers are satisfied with these triples and seek no further complications. The URI prefixes in the paper are listed in Listing 1 [6].

```
:ex1 a chess-o:ChessGame ;
    chess-o:hasWhitePlayerName "Bobby Fischer" .
```

Listing 2. Dataset A: two simple RDF triples may constitute a view

However, one might think of a case where details on the person of Bobby Fischer are needed, e.g., as provided by Dataset B in Listing 3.

```
:ex1 a chess-o:ChessGame ;
  chess-o:hasWhitePlayer [
     a chess-o:Agent ;
     chess-o:hasName "Bobby Fischer" ;
     chess-o:hasELORating "2780" ;
     skos:closeMatch <http://dbpedia.org/resource/Bobby_Fischer> .
     ] .
```

Listing 3. Dataset B: data conforming to the pattern in Listing 4

As shown in Figure 1, Dataset B corresponds to a slightly more complex graph structure than Dataset A, possessing an additional resource node that Dataset A lacks. Going one step further, Dataset B may be actually be published according to an ODP $\mathcal{O}$, given in Listing 4, where it is stated that chess games are played by exactly one agent as white player.

[6] At the submission time the URI is `http://salonica.dia.fi.upm.es:8080/rdfchess/`

```
chess-o:hasWhitePlayer rdf:type owl:ObjectProperty .
chess-o:Agent rdf:type owl:Class .
chess-o:ChessGame rdf:type owl:Class ;
  rdfs:subClassOf [ rdf:type owl:Restriction ;
     owl:onProperty chess-o:hasWhitePlayer ;
     owl:onClass :Agent ;
     owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger
  ] .
```

Listing 4: Ontology Design Pattern $\mathcal{O}$ used by Dataset B

The benefit of the more complex structure for Dataset B is the richer information content and the flexibility for even more information about the agent who was the white player of the chess game, hence increasing the potential for reuse. On the other hand, publishing the dataset under the form of Dataset A may satisfy certain audience expecting simplicity. Despite this, we can still see that both datasets still contain roughly the same information regarding the name of the white player in the chess game. In this context, in addition to saying that the Dataset B is published according to the *ontology design pattern* $\mathcal{O}$, we may also say that the Dataset A is published according to a *view* for $\mathcal{O}$, which simply contains two signature declarations (a class and an object property).

More precisely, both ODPs and views for ODPs are ontologies that can act as schemas over data – henceforth, we use the term ontology and (linked data) schema interchangeably. An ontology is seen as an ODP by virtue of qualitative characteristics, such as being well-engineered, concise, able to cater multiple perspectives and granularity, highly reusable, modular, and highly focused on modeling only a single key notion in a domain. Moreover, existing literature hardly mentioned a crisp, formal mathematical definition to distinguish ODPs from ordinary ontologies. Despite this, we do not intend to present a formal, mathematical definition of ODPs, but rather rely on the assumption that ontologies possessing the aforementioned characteristics do exist online. In fact, there are known online repositories[7] that host such ontologies and call them ODPs. If an ODP contains abstractions that are typically designed to cater to multiple perspectives, a view of the ODP should be understood as a simplified form of the ODP for a *particular* perspective from some data provider or consumer. Consequently, an ODP can have multiple views, and some bridges are needed between ODPs and its views to allow them to work together. These bridges are realized in the form of SPARQL queries, which enable a rewriting from data according to the ODP into data according to the view, and vice versa.

2.2 Pattern Flattening and View Expansion

In the example before, we contend that a data publisher may want to maintain Dataset B while publishing Dataset A, and that transformations from Dataset A to B and vice versa are possible in some cases. We may name these operations as *view expansion* and *pattern flattening* respectively. For the example above,

[7] E.g.,`http://ontologydesignpatterns.org` and `http://www.gong.manchester.ac.uk/odp/html`

Query 1 in Listing 5 is an example of view expansion, while Query 2 in Figure Listing 6 shows the counterpart pattern flattening. Query 1 *expands* the data about the white player of a chess game from one triple that gives us only the name of the player as string into a set of triples with a URI resource representing a player, in addition to his name as string. Meanwhile, Query 2 *flattens* the data since it does the opposite of Query 1. In a possible scenario, Query 1 would be launched by a non-expert data publisher upon arrival of extended information.

```
DELETE { chess:game123 chess-o:hasWhitePlayerName "Bobby Fischer" . }
INSERT {
 chess:game123 chess-o:hasWhitePlayer chess:player123 .
 chess:player123 chess-o:hasName "Bobby Fischer" .
 chess:player123 skos:closeMatch <http://dbpedia.org/resource/Bobby_Fischer> .
}
WHERE { chess:game123 chess-o:hasWhitePlayerName "Bobby Fischer" }
```

Listing 5. Query 1: view expansion example

```
DELETE {
  ?chessplayer ?p ?o .
  ?game chess-o:hasWhitePlayer ?chessplayer .
  }
INSERT {  ?game chess-o:hasWhitePlayerName ?name .  }
WHERE {
  ?chessplayer ?p ?o .
  ?game chess-o:hasWhitePlayer ?chessplayer .
  ?chessplayer chess-o:hasName ?name .
}
```

Listing 6. Query 2: pattern flattening example

The ontology pattern and the view for the previous example, based on which Query 1 and 2 were formulated, are shown together in Figure 1. In our example, when modeling the notion of chess game as an ontology or a content pattern, it is likely that the structure resembling Dataset B is favored since it would allow one to specify more precisely the definition of "chess game". On the other hand, linked data publishers often dislike complex structures because publishing linked data from data sources such as relational databases or spreadsheets to such structures is presumably not so straightforward. Now, by defining view expansion and pattern flattening such as Query 1 and 2, we can reconcile these two seemingly opposite points of view. This linked data publication style benefits from well-designed patterns, but can still be simple enough for many linked data publishers.

2.3 Shapes Constraint Language for Checking Pattern Conformance

The SHACL (Shapes Constraint Language) is a RDF vocabulary for describing RDF graph structures [12], still under specification by the W3C RDF Data

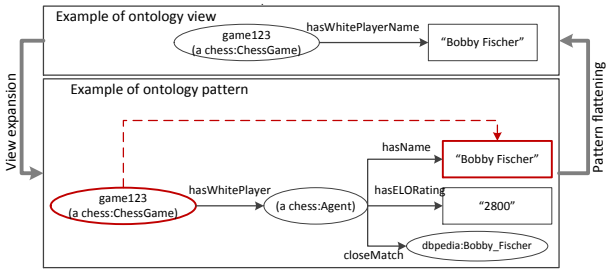

Fig. 1: Triples in Dataset A (top part) can be obtained from Dataset B (bottom part) by introducing a shortcut represented with the dotted line. Ellipses represent class instances and rectangles literals. Arrows represent object or datatype properties.

Shapes Working Group[8]. RDF shapes declare constraints about RDF nodes in terms of expected cardinalities, datatypes and other restrictions. By declaring shape expressions, the validation of data structures is possible (i.e. one chess game as RDF), the expected graph patterns can be better communicated and interoperability is better realized.

We contend that RDF Shapes should be used to grant the conformance of a data structure to a certain pattern, and that ODPs should include in their description the corresponding shapes that validate data structures. The pattern in Listing 4 can be restricted by the RDF shape in Listing 7.

```
chess-o:ChessGame a sh:Shape ;
    sh:property [
        a sh:PropertyConstraint ;
        rdfs:comment "The chess white player" ;
        sh:predicate chess-o:hasWhitePlayer ;
        sh:valueType chess-o:Agent ;
        sh:cardinality 1
    ] .
```

Listing 7. RDF Shape for the pattern in Listing 4

While the restrictions presented in the example are the same as those in Listing 4, the intention is different as validation is done under a closed-world assumption. The same approach had been taken by the former SPIN specification [11], for which stable and high quality implementations exist, as well as by other constraints specifications like RDF Unit, OWL Constraints or Stardog ICV. Here, validation is driven by constraints, which are separate from classes, and a validation function takes two arguments: the RDF graph with the constratins the RDF datasets with the data to be validate. The validation can in many cases be presented as a SPARQL query –directly supported by SHACL and by SPIN. An example of validating query is presented in Listing 8.

[8] http://www.w3.org/2014/data-shapes

```
[ rdf:type sh:Constraint ;
 sh:severity sh:fatalError ;
 sh:report "SELECT ?this" ;
 sh:classScope "http://purl.org/NET/ChessGame"^^xsd:anyURI ;
 sh:sparqlShape
   """FILTER NOT EXISTS { ?this chess-o:hasWhitePlayer ?v .
        FILTER NOT EXISTS { ?v rdf:type chess-o:Agent . } }"""
] .
```

Listing 8. Validation of a shape constraint

3 Advantages of pattern-based linked data publication

The advantages which could be obtained if a significant amount of linked data publishing were based on ontology design patterns with validation mechanisms include:

Higher quality of documentation of underlying graph structures. If patterns are reused, then the effort of documenting this patterns becomes more worth while, and can also become a collaborative effort. One pattern, well documented, can potentially be used by many dataset publishers. This multiplier effect thus gives higher return of investment for providing high quality documentation.

Improved understandability of linked datasets by data engineers. Once patterns become established for widespread use, data engineers will easily understand and recognize them, which will help in understanding other parties' datasets.

Less effort needed in integrating and federated querying of linked datasets. Improved understandability of datasets will simplify reuse: Data engineers will not have to invest as much time and effort in understanding underlying graph structures and the intentions behind it, as they will already be familiar with the patterns. Thus integration of data, e.g. for federated querying, is simplified.

Avoidance of the complications of large ontologies as underlying schema. While some of the benefits mentioned above could also be reaped from adopting large, standardized, ontologies, the community seems rather painfully aware of the fact that this is not a viable approach, for several reasons, e.g. because large ontologies are often not adequate for publishing a new dataset due to incompatible ontological commitments, due to overly complicated schema, or due to a misfit in granularity of representation. Using ontology design patterns avoids these problems, since patterns are small and modularly composable, and as such are also easily modifiable (while some key recognizable parts can still be retained).

Standardization of graph patterns will not be necessary; however standardization of core patterns is still possible. Patterns are small and easy to reuse, and thus they lend themselves easily to social processes which can establish de-facto standards, without the need of central control. At the same time, if the community deems it helpful, some central patterns could also be standardized, e.g. through some of the common standardization bodies.

OWL axiomatizations for reasoning available if needed. A well-designed pattern usually comes with an OWL axiomatization, the main purpose of which is to disambiguate the meaning of the pattern for human users of the pattern. At the same time, however, the axioms can be repurposed for automated reasoning if this is helpful, e.g. for more sophisticated query answering.

4 The Linked Chess Dataset

The ideas of this paper are illustrated with a chess dataset published online[9], which is of interest per se: since the early chess sites online many chess databases have been offered for free[10] but to the knowledge of the authors the dataset presented here is the only one published as linked data and linked to other datasets.[11]

4.1 Chess games and the PGN format

Standard chess games can be represented as sequences of moves together with some metadata information. Most of games on the web are published in the Portable Game Notation (PGN) format. PGN was developed in 1994 with the intent to 'facilitate the sharing of public domain chess game data among chess players (both organic and otherwise), publishers, and computer chess researchers.'[12] A sample PGN record (taken from the specification) is shown below.

```
[Event "F/S Return Match"]
[Site "Belgrade, Serbia JUG"]
[Date "1992.11.04"]
[Round "29"]
[White "Fischer, Robert J."]
[Black "Spassky, Boris V."]
[Result "1/2-1/2"]

1. e4 e5 2. Nf3 Nc6 3. Bb5 a6 4. Ba4 Nf6 5. O-O Be7 6. Re1 b5
7. Bb3 d6 8. c3 O-O 9. h3 Nb8 10. d4 Nbd7 11. c4 c6 12. cxb5
[moves abridged] 41. Ra6 Nf2 42. g4 Bd3 43. Re6 1/2-1/2
```

The information in square brackets is metadata about the match, including the players involved, when and where it was played, and the result. The remaining information, in the numbered list at the bottom, is the sequence of moves made during the game, represented in Standard Algebraic Notation (SAN). Each of the squares in the board is identified by a letter [a-h] and a number [1-8], and SAN moves describe implicitly or not which is the origin of the piece to be moved and which is the destination. For instance, 1. e4 e5 indicates that the white player moved his pawn to e4 and black countered by moving his own pawn to e5. PGN files are not optimal regarding chess players' identification, the precise location of

[9] http://salonica.dia.fi.upm.es:8080/rdfchess

[10] This is the case of database.chessbase.com, chessgames.com, www.365chess.com or www.chessbites.com, each of them with a few millions of games.

[11] The pattern is to appear as 'An Ontology Design Pattern for Chess Games', by A. Krisnadhi et al. in the Proc. of the 6th Int. W. on Ontology Patterns proc. (2015)

[12] http://www6.chessclub.com/help/PGN-spec

the events or other context information. For example, different annotators may represent players' names differently, e.g. by using initials or nicknames, and they may use different mechanisms for expressing the location, such as latitude and longitude instead of place name. A linked data representation allows the data to be expressed in less ambiguous terms by providing links to URIs representing the precise entities involved.

4.2 The Linked Chess Dataset

The Linked Chess Dataset consists of chessplayers, openings and chess games served as Linked Data and accessible from a SPARQL endpoint[13]. Resources are dereferenceable and served with content negotiation, the HTML page being enriched with information from the linked resources. An implementation of the view expansion and pattern flattening operations is also offered, as well as a service that transforms a game in PGN into RDF. A description of the data models is also included in the website. URIs of these resources were chosen following the format `DOMAIN/resource/CLASS/ID'` where CLASS is one of `{chessplayer,chessgame,opening,location}`, and ID is systematically formed for openings (ECO) and players (name+surname) but formed randomly for chessgames.

Chess games were harvested from the web using a customized web crawler[14]. After discarding duplicated games, the collected PGN files were transformed to the RDF data structure in its simplest form (the *view*). In the publication process, comments were dropped, as they could be object of copyright protection –while in general chess games are not. [15]. The views were then expanded into the full pattern model whenever additional information was available: (i) locations were guessed from 'geonames.org' using the geonames API[16], with high accuracy results; (ii) chessplayers were linked to those in dbpedia and freebase by using the Spotlight API[17] – as of June 2015, the number of chessplayer in dbpedia amounted 1220; (iii) chess openings were linked to dbpedia resources by matching the ECO[18] code. In addition, a number of openings was matched to

[13] `http://salonica.dia.fi.upm.es:3030/RDFChess/query`

[14] While it would have been feasible to use a standard web crawler such as HTTrack, Nutch or Crawler4j we elected to write our own to take advantage of the constrained nature of this task

[15] Chess games satisfy the requirements to qualify as objects of intellectual property rights because they are products of the human mind, they are attributable to a person or persons and they are fixated in a tangible medium. However, courts do not deem them as copyrightable. See 'Copyright on Chess Games', Edward Winter, 1987, online at `http://www.chesshistory.com/winter/extra/copyright.html`

[16] `http://www.geonames.org/source-code/javadoc/`

[17] `http://spotlight.dbpedia.org`

[18] ECO stands for *Encyclopaedia of Chess Openings*. ECO codes consist of a letter (from 'A' to 'E') plus a sequence of two numbers; ECO codes classify chess openings in a hierarchical manner. Thus, the example game in Section 3.1 executes the opening C95, which corresponds to the Spanish opening, variant Morphy Defense.

the corresponding entities in the terminology used by the Library of Congress[19] and hierarchy relations between openings was added.

4.3 The Linked Data Chess example

According to the simplest possible view, the metadata of the sample game in Section 4 can be represented as in Listing 9. Rounds and tournament information are appended using the `sem:subEventOf` property and each round is modeled as a sub-event of a chess competition in the same manner. The chess moves are represented as a linked list of half-moves (not shown in the listing).

```
game:ccfb9754-30ec-4a55-8f18-adebe9db9071
        a                           chess-o:ChessGame ;
        chess-o:atTime              "1992.??.??" ;
        chess-o:hasBlackPlayerName  "Fischer, Robert James" ;
        chess-o:hasChessGameAtNamedPlace "Belgrade" ;
        chess-o:hasECOOpening       "E83" ;
        chess-o:hasPGNResult        "0-1" ;
        chess-o:hasWhitePlayerName  "Spassky, Boris V" ;
```

Listing 9. Chess game metadata as a view

The simplest view expansion for the chess opening adheres to a pattern – replacing a literal by a typed resource but adding no information (Listing 10).

```
game:ccfb9754-30ec-4a55-8f18-adebe9db9071
 chess-o:hasChessGameOpening opening:E83 .
opening:E83
   a chess-o:ChessGameOpening ;
 chess-o:ECOID "E83" .
```

Listing 10. Simplest view expansion

However, this information is suitable to be fully expanded, as in Listing 11.

```
opening:E83
   a   chess-o:ChessGameOpening ;
   rdfs:label  "King's Indian, Samisch" ;
   chess-o:ECOID "E83" ;
   skos:broaderTransitive opening:E84 ;
   skos:closeMatch <http://id.loc.gov/authorities/subjects/sh00008710>
  , <http://dbpedia.org/resource/King's_Indian_Defence> ;
   skos:narrowerTransitive opening:E81 .
```

Listing 11. Full view expansion

[19] The Spanish Opening corresponds to the Library of Congress Subject Headings URI `http://id.loc.gov/authorities/subjects/sh98003603`, which enriches our information on the opening with alternative names, etc.

The hierarchical classification of chess openings permits adding the 'parent' opening (E81) and a descendant line (E84 Panno Main line). External algorithms add the links to other datasets (subjects of the Library of Congress and dbpedia) and an English name for the variant. Similar expansions can be made with location, if (and only if) it has been matched to a known place in GeoNames (see Listing 12).

```
location:792680  a  chess-o:Place ;
        chess-o:hasName  "Belgrade" ;
        skos:closeMatch  <http://sws.geonames.org/792680> .
```

Listing 12. Data conforming to the class Place

Chess players and chess events can be expanded in a similar manner on demand, adjusting the expansion to the actual availability of extra data. The full expanded version of the view in Listing 9 is given in Listing 13. The complete SPARQL queries to expand the views or to flatten the patterns, as well as some validating RDF shapes are provided in the website.

```
game:ccfb9754-30ec-4a55-8f18-adebe9db9071
   a   chess-o:ChessGame ;
   chess-o:atPlace         location:792680 ;
   chess-o:atTime          "1992.11.04" ;
   chess-o:hasPGNResult    "1/2-1/2" ;
   chess-o:hasBlackPlayer  player:Bobby+Fischer ;
   chess-o:hasChessGameOpening opening:E83 ;
   chess-o:hasPGNResult    "0-1" ;
   chess-o:hasWhitePlayer  player:Boris+Spassky .
```

Listing 13. Fully expanded view

5 Conclusions

This work has contributed to bridge the visions from ontologists and linked data publishers by proposing the use of ontology design patterns together with the SPARQL queries to achieve simplified views thereof and constraints to validate the adherence of data structures to the patterns. In addition, a relevant dataset of chess has been released, demonstrating these concepts and also providing a reference point for chess practitioners. The six advantages claimed in Section 3 cannot be easily evaluated, as they belong to a publication style not yet fully materialized. However, an interesting study would consider how much of the linked data online is based on pattern-based ontologies, or how much of the data is constrained by SPIN or SHACL –this remains as future work. [20]

[20] **Acknowledgements**. This work has been partially supported by the NSF under awards 1440202 *EarthCube Building Blocks: Collaborative Proposal: GeoLink - Leveraging Semantics and*

References

1. Eva Blomqvist and Kurt Sandkuhl. Patterns in ontology engineering: Classification of ontology patterns. In Chin-Sheng Chen et al., editor, *ICEIS Proc. of the 7th Int. Conf. on Enterprise Information System*, pages 413–416, 2005.
2. Oscar Corcho, María Poveda-Villalón, and Asunción Gómez-Pérez. Ontology engineering in the era of linked data. *Bulletin of the Information Association for the Information Age*, April 2015.
3. Aldo Gangemi. Ontology design patterns for semantic web content. In Yolanda Gil et al., editor, *Proc. of the 4th Int. Semantic Web Conference, ISWC 2005*, volume 3729 of *LNCS*, pages 262–276. Springer, 2005.
4. Asunción Gómez-Pérez, Mariano Fernández-López, and Óscar Corcho. *Ontological Engineering: With Examples from the Areas of Knowledge Management, e-Commerce and the Semantic Web*. Springer-Verlag, Secaucus, NJ, USA, 2007.
5. Thomas R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. *Int. J. Hum.-Comput. Stud.*, 43(5-6):907–928, December 1995.
6. Harry Halpin, Patrick J. Hayes, James P. McCusker, Deborah L. McGuinness, and Henry S. Thompson. When owl:sameAs isn't the same: An analysis of identity in linked data. In Peter F. Patel-Schneider et al., editor, *Proc. of the 9th International Semantic Web Conference*, volume 6496 of *LNCS*, pages 305–320. Springer, 2010.
7. Prateek Jain, Pascal Hitzler, Peter Z. Yeh, Kunal Verma, and Amit P. Sheth. Linked data is merely more data. In *Linked Data Meets Artificial Intelligence, Papers from the 2010 AAAI Spring Symposium, Technical Report SS-10-07, Stanford, California, USA, March 22-24, 2010*. AAAI, 2010.
8. Krzysztof Janowicz. Place and location on the web of linked data. http://stko.geog.ucsb.edu/location_linked_data, Nov. 2012. Retrieved Jun. 9, 2015.
9. Krzysztof Janowicz, Pascal Hitzler, Benjamin Adams, Dave Kolas, and Charles Vardeman. Five stars of linked data vocabulary use. *Semantic Web*, 5(3):173–176, 2014.
10. Amit Krishna Joshi, Prateek Jain, Pascal Hitzler, Peter Z. Yeh, Kunal Verma, Amit P. Sheth, and Mariana Damova. Alignment-based querying of linked open data. In Robert Meersman et al., editor, *Proc. of On the Move to Meaningful Internet Systems: OTM 2012, Part II*, volume 7566 of *LNCS*, pages 807–824. Springer, 2012.
11. Holger Knublauch. SPIN - Modeling Vocabulary. Technical report, W3C Member Submission 22 February 2011, 2011.
12. Eric Prud'hommeaux, José Emilio Labra Gayo, and Harold R. Solbrig. Shape expressions: an RDF validation and transformation language. In *Proc. of the 10th Int. Conf. on Semantic Systems, SEMANTICS*, pages 32–40, 2014.
13. W. V. Quine. On what there is. In Tim Crane and Katalin Farkas, editors, *From a Logical Point of View*, volume 2, pages 21–38. Harvard University Press, 1961.

Linked Data for Data Sharing and Discovery in the Geosciences, by the EC under the International Reseach Staff Exchange Scheme (IRSES) of the EU Marie Curie Actions, project *SemData – Semantic Data Management*, and by the Spanish Ministry of Economy and Competitiveness (project TIN2013-46238-C4-2-R).