Chapter 11

Modeling Ontology Design Patterns with Domain Experts – A View From the Trenches

Krzysztof Janowicz, University of California, Santa Barbara, USA

11.1. Introduction and Motivation

Ontology engineering methods, frameworks, and tools have been studied for more than 20 years and have been widely applied in academia, industry, and at government agencies. While there are considerable differences, most established methods share a common core, namely a knowledge acquisition activity that spans over phases such as (requirements) specification, conceptualization, and formalization with the goal of effectively eliciting knowledge from domain experts to implement an ontology [4, 10, 15, 19–22]. Despite substantial progress, knowledge acquisition remains a key bottleneck and has been extensively studied in work on expert systems since the 1980s [14]. Some of the underlying problems such as scaling can be addressed by a combination of classical top-down engineering with bottom-up, data-driven techniques [5]. Today, this is increasingly important for arriving at a deeper axiomatization of Linked Data to improve interoperability and query federation.

However, one can also look at the problem of acquiring knowledge from domain experts from an entirely different angle and argue that the challenge lies in reaching agreement between domain experts [12] together with the misconception that there has to be a common understanding of terminology within a particular domain. While there should be a broad and domain overarching agreement on units of measure, the same cannot be said about concepts such as *environment*, *resilience*, *forest*, *city*, and so forth. This is not only true across domains but even within research fields and communities [10]. This misconception may have its roots in the successful standardization of protocols, formats, and languages which may have caused ontology engineers and domain experts to believe that one could and should also *standardize meaning* on the Web.

The meaning of terms, however, depends on different viewpoints, cultures, underlying themes, legislation, state of knowledge, level of detail, current context, and many other aspects that jointly lead to the vast heterogeneity of data published on the Web today. To give an intuitive and commonly used example, a street can be conceptualized as a *connection* between two places from the view point of transportation science or as a *separation* that cuts a habitat into pieces from the view point of ecology. This heterogeneity is not a problem that needs to be *resolved* but a resource that enables us to address challenges that cannot be fully understood by considering one of these aspects in isolation. Nonetheless, some common core is needed to retrieve, reuse, and integrate data across different sources. In other words, communication and thus interoperability break down [17] if there is no overlap between the underlying conceptualizations of the shared (physical) space. As *specifications* of these conceptualizations, ontologies provide us with the means to approximate commonalities and differences between datasets, communities, domains, and so forth. Simply put, the purpose of ontologies is to make the intended meaning *explicit*, not to *agree* on what terms mean.

Following this argumentation, the key challenge for ontology engineering becomes striking a balance between fostering interoperability without restricting semantic heterogeneity, i.e., without enforcing a common understanding. One approach is to defer the introduction of concepts that are heavy on *ontological commitments*, e.g., *vulnerability*, and first focus on the development, reuse, and combination of common building blocks [10]. The resulting ontologies reflect the conceptualizations of the data providers and still ensure a minimum fallback level for interoperability via the usage of these common building blocks. Content ontology design patterns [6] are a promising candidate for such an approach and also act as a middle ground between engineering local ontologies from scratch and trying to reuse existing ontologies despite their differing ontological commitments [2, 16].

In this work, we describe our experience in modeling such content patterns with domain experts from a variety of different domains and fields including geography, oceanography, geology, industrial ecology, transportation science, architecture, the digital humanities, and so forth. The presented work reflects on the lessons learned and challenges of running more than a dozen vocabulary camps (VoCamps) and similar events over the past six years. While we will focus on multi-day vocabulary camps here, the discussion can be generalized to other ontology engineering events that bring domain experts and ontology engineers together.

In the following, we will address the *why, what, and how* of pattern modeling with domain exerts. First, we will briefly revisit the *perceived* value proposition of ontologies and Semantic Web technologies for domain experts to better understand their expectations and reasons for participating in the development of ontologies. We will then discuss how VoCamps are structured and why we believe they are a successful approach to pattern engineering. Finally, using the Semantic Trajectory pattern [8], we will outline modeling choices by example and highlight how they foster reusability and flexibility of patterns.

11.2. The Value Proposition of Ontologies

In the following we will revisit the value proposition of ontologies from the view point of individual domain experts (here researchers), in contrast to large-scale data providers, government agencies, industry, and so forth, to better understand *why* they become interested in ontologies and participate in vocabulary camps. We will structure the value proposition into three different stages: publishing and retrieving data, interacting with data, and reusing and integrating data. Note that in contrast to previous work [11], we focus on the *perceived* value proposition.

11.2.1. Publishing and Retrieving Data

The added value of ontologies starts with publishing own data. There are at least three different motives to semantically annotate these data by means of ontologies. First, researchers hope to improve the discoverability of their scientific results beyond mere keyword search. This aspect goes hand in hand with the hope to be able to better and faster retrieve useful data from others. As a consequence, many domain experts approach ontology engineering from the perspective of building richer taxonomies. Second, an increasing number of funding organizations, scientific journals, and universities require data publication and managing strategies. Semantic Web technologies are perceived as a promising choice for doing so. The thirds and final reason is less obvious despite having the most important long-term effects. By semantically annotating their data, scientists improve the reproducibility of their results, e.g., by adding provenance records, and reduce the risk of (accidental) misinterpretation and therefore wrong usage of their own data [18]. Based on our observations, many domain experts interested in ontology engineering have worked with databases and conceptual modeling environments before and thus may falsely expect that ontologies are primarily used for integrity constrains.

11.2.2. Interacting with Data

Knowledge exploration, e.g., by follow-your-nose browsing or faceted search, are common ways to interact with Linked Data and in many cases more attractive and intuitive to domain experts than querying endpoints using SPARQL. A common hope associated with Linked Data is that the available data hubs will store different information about common entities and thus enable a more holistic view. What is often overlooked, however, is the fact that these data sources will likely contain overlapping information. To give an example, there will be multiple population counts for a populated place and multiple geographic coordinates for its centroid. Given the decentralized nature of the Web and current research focus, a majority of exciting frameworks do not yet consider Linked Data fusion/conflation [1]. Consequently, the perceived value proposition of Semantic Web technologies differs from the research focus. From an ontology modeling perspective, one could argue that a stronger axiomatic foundation will ease Linked Data fusion, e.g., by identifying functional properties. For now, Linked Data users have to handle contradicting data themselves.

11.2.3. Reusing and Integrating Data

Discovering and interacting with data on the Web is often a means to an end with the reuse and integration of the data being the final goal. A very common misconception among domain experts is the believe that the creation of Linked Data and here especially the use of owl:SameAs will magically enable query federation. The realization that even a densely linked global graph of data does not necessarily enable queries over multiple of the involved data hubs often comes as a shock [9]. Understandably, the first reaction is to call for common top-level and domain ontologies to ensure that the same classes and properties are used across these data sources. It takes a deeper understanding along the lines of argumentation made in the introduction section and a basic understanding of ontology *alignment* to realize why domain ontologies do not exist for the vast majority of research fields and why those that exist are facing substantial challenges. A second misconception exists with respect to the formal semantics of knowledge representation languages such as OWL. As discussed above, many domain experts and data providers approach ontologies from an integrity constraints or object-oriented design perspective and do not fully understand the consequences of an inferential semantics and the Open World Assumption. The most common example are (global) domain and range restrictions and the believe that they would constrain the usage of properties. This is one of the reasons why vocabulary camps make use of *guarded* domains and range restrictions instead [13].

11.3. Vocabulary Camps for Pattern Engineering

In this section we will introduce VoCamps as a means to develop content patterns and outline *how* they are set up. Most of what will be said can be generalized to related forms of synchronous knowledge acquisition.

11.3.1. General Setup and Scope

The events organized over the past years were scheduled for 2-3 days. Here we will outline a 2-day event. VoCamps are *unconferences*, i.e., there is no registration fee, no formal presentations (aside of selected invited talks), no proceedings, and so forth. Instead VoCamps focus on bringing domain experts and ontology engineers together to address real modeling issues. (Geo)VoCamps typically draw between 20 and 40 attendees. While the composition varies substantially, arriving at a balanced combination of ontology engineers and domain experts was not a problem so far. Ideally there would be one experienced ontology engineer per 2-3 domain experts. At least in case of GeoVoCamps, a team of 6-8 regular participants ensures continuity and brings in the required expertise.

VoCamps typically start with domain experts that bring their modeling problems to the event. From these, 2-3 are selected and addressed during the VoCamp. The key success to a productive event is to have relatively short but intense breakout sessions with frequent reports back to uncover similarities and differences with the other groups/patterns and to identify potential issues and improvements. The

Table 11.1. A representative agenda for a VoCamp.

	1st day	2nd day
Morning	• Introduction • Report from previous VoCamps • A selected pattern as example • Invited talks	• Brief recap • Breakout groups work on patterns • Reports from the groups • Breakout groups (*implementation*)
Afternoon	• Decide on breakout groups • Breakout groups work on patterns • Reports from the groups	• Reports from the groups • Breakout groups (*examples/data*) • Reports from the groups • Brief documentation

goal of a VoCamp is typically the conceptualization and draft axiomatization of a pattern as well as a brief documentation together with examples. In the weeks following a VoCamp, the results are improved and polished and finally published as an (OWL) ontology together with a paper documenting and motivating the work. Follow-up VoCamps then take up on this work to align their own patterns, learn from the modeling and implementation approaches taken before, or refine certain aspects of the pattern. Experience from the last years show that at least 2 of 3 patterns started at a VoCamp are eventually published.

It is too early to determine the long-term impact of the developed patterns and the VoCamp model.[1] However, some of them, e.g., the trajectory or agent-role patterns, have been used in various settings. At this stage, we believe that bringing domain experts and ontology engineers together and spreading an understanding for the possibilities and limitations of Semantic Web technologies is the core outcome. The uptake of ontologies (and patterns more specifically) in large cyber-infrastructures such as NSF's Earthcube as well as a growing interest among domain experts from multiple disciplines provides positive signals.

11.3.2. VoCamp Agenda and Workflow

Table 11.1 outlines a representative agenda of a VoCamp. The morning of the first day is used to introduce the unconference-style to new participants, to report on previous VoCamps and their results thereby ensuring continuity, as well as a brief presentation outlining the design and implementation of a pattern as example. These three items set the stage and scope for the event. This is typically followed by one or two invited short talks of about 20 minutes that introduce an interesting domain or research field or provide a novel perspective on patterns and ontology engineering. Finally, domain experts get a chance to pitch their modeling projects and ideas before the lunch break.

The afternoon of the first day usually begins with selecting 2-4 potential topics/patterns and then splitting into breakout groups. Ideally, each group consists of 6-10 members with at least one experienced ontology engineers, domain expert, data provider, and dedicated scribe (or moderator). During the first breakout session which takes 2-3 hours, each group decides on the scope of their work, i.e.,

---

[1] See also `http://dase.cs.wright.edu/blog/geovocamps-taking-stock`.

where to start and stop modeling a certain topic or domain problem. This phase is also characterized by discussions on the used domain terminology and is the most intense part of the VoCamp as no common ground has been established so far. Explaining that multiple views can be reconciled and that different modeling choices can be transfered into each other often helps to prevent *turf wars* between domain experts. That said, different and contradicting views points brought in by domain experts are the most fruitful aspect of this scoping phase as they will later ensure the reusability and extensibility of the developed pattern. The first afternoon ends with a reporting and feedback session. These sessions are of fundamental importance as allowing members from other groups to actively comment on and contribute to other patterns helps to establish a constructive atmosphere and ensures *buy-in* for the final patterns. Finally, the participants also discuss reoccurring modeling choices and strategies which will further ease the later alignment of the patterns.

The second morning and afternoon sessions follow the same template of intense work within breakout groups and frequent reports. The most important difference, however, is that a draft implementation of the patterns (typically in OWL) has to be developed and presented by the end of the second morning. The afternoon breakout session is then used to populate the pattern with real data and to document the work to a degree where it can be finished in an asynchronous style as the participants come from different institutions and countries.

There is no specific ontology engineering method that is used across all groups and VoCamps. Nonetheless, there are a few common aspects that have emerges as successful strategies. One of them is the formulation of competence questions [7] during the first breakout session and the use of tools to develop concept maps on a shared screen. Working with real data, populating the draft patterns, and writing SPARQL queries to retrieve the data are other commonly used strategies. Finally, some of the core VoCamp members rotate between groups to keep the meeting productive and to ensure that no single domain expert or ontology engineer dominates the discussion.

11.4. Pattern Design Decisions by Example

A discussion of what distinguishes a pattern from a small ontology and how to best engineer reusable patterns is out of scope here. Nonetheless, it is worth looking at a specific example of a pattern designed at a VoCamp to understand why certain choices have been made, how they enable a broad and domain spanning usage of the pattern, and *what* kinds of problems are typically addressed. We will use the Semantic Trajectory pattern as example [8]. More specifically, we will argue how the pattern fulfills the criteria for patterns informally defined as quality and reusability proxies at VoCamps, namely that patterns should:

- Cover a wide range of domains or application areas.
- Be extensible to provide additional details.
- Supports multiple granularities.
- Provide an axiomatization beyond mere surface semantics.
- Have various *hooks* to well-known ontologies / patterns.

- Be self-contained to a degree where they can be used on their own.

It is interesting to note that the second and last point require striking the right balance between developing a pattern that is generic enough to act as a building block for an application ontology but self-contained (and specific) to a degree where a meaningful use, e.g., the creation of Linked Data, does not require additional ontologies. The deeper axiomatization of the pattern is left to a later chapter in this volume.

Trajectories play a key role in many areas, be it for wildlife tracking, studies of human movement, traffic analysis, scene modeling, cruises of oceanographic research vessels, health and exposure research, to name but a few. Simply put, a trajectory is the path that a moving object takes through space as a function of time. While such definition also includes the trajectory of a projectile, the pattern is designed for cases that benefit from a semantic annotation of the moving object, the places visited, the mode of transportation, the sensor used to determine the objects location at a given time, and so forth, thus forming a *semantic* trajectory.

To do so, the pattern introduces classes such as *fix*, *segment*, *moving object*, *position*, *source*, and *attribute*, together with relations between them. Fixes, for instance, determine the position of an object at a certain time and are observed by some source. Two successive fixes of the same trajectory are connected by a segment, i.e., a linear interpolation of the path taken. The segments themselves are traversed by a moving object.

Interestingly, as depicted in figure 11.1, this minimal pattern already supports a wide range of use cases. For instance, one could model the path taken from UCSB to the Los Angeles airport as a single segment connecting the start fix Santa Barbara with the end fix Los Angeles. In such case, the fixes are not just arbitrary measurements taken by the used positioning technology, e.g., a GPS-based navigation system, based on the device's sampling rate but meaningful *places*. In this case, the segment represents U.S. Highway 101 and the moving object (if given) could be a car or bus. Such modeling, of course, does not imply that the highway is indeed a linear feature nor that only two fixes were taken. Instead, it reflects the needs, here the resolution, of an application or use case. A typical example would be studying origin-destination trips. On the other extreme, one can approximate the exact path taken by a moving object by increasing the number of fixes and thus shortening the linear segments up to the level supported by the used positioning technology. For example, a modern smartphone can take GPS fixes once every second with an accuracy of about 5 meters. Here a typical use case for the pattern would be recreational hiking. Finally, the pattern can also cover the middle ground. In the first example all fixes were places, while they were mere positioning artifacts in the second case. In many trip planning applications or wildlife monitoring it is important to roughly approximate the path taken and also to identify certain Points Of Interest (POI). Intuitively a fix taken at a fuel station or watering place carries additional meaning. Summing up, the Semantic Trajectory pattern supports multiple granularities and can be used for a wide range of use cases. It is also self-contained in the sense that the described examples can be fully modeled with the pattern alone.

The pattern also provides multiple opportunities for integration with other well established patterns and ontologies. For instance, the pattern specifies that

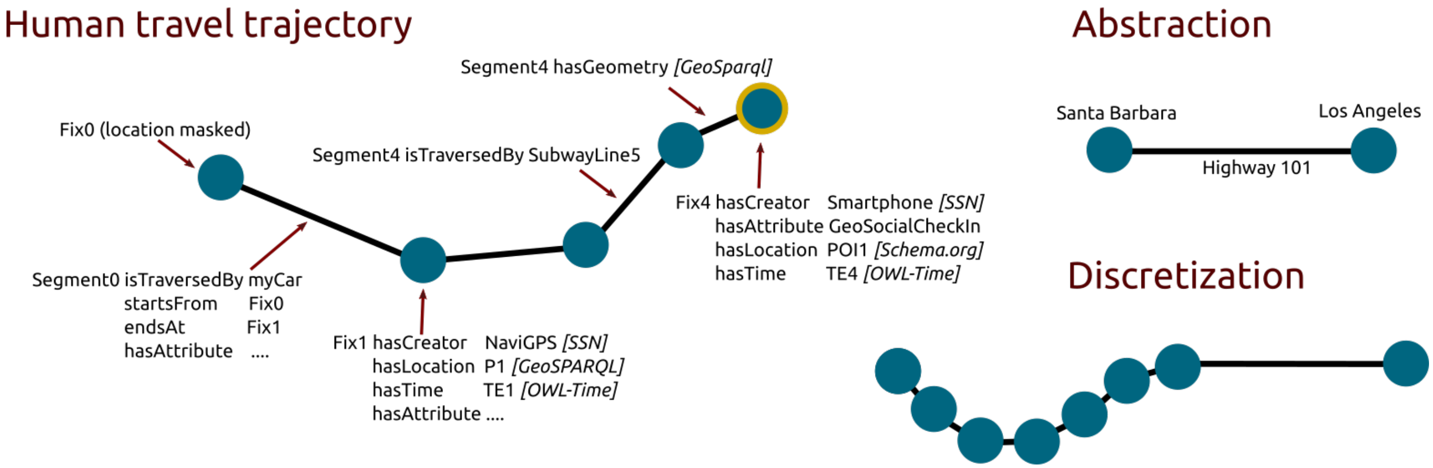

Figure 11.1. Flexible modeling via the Semantic Trajectory pattern.

fixes are created by some *source* but does not provide any further axiomatization for this source. Consequently, the source could be a GPS sensor, a human observer, or even a written itinerary from a historical expedition. If the source is indeed a sensor, it can be further specified, e.g., to describe the frequency (*ssn:Frequency*) using the W3C-XG Semantic Sensor Network ontology (SSN) [3]. Similarly, the pattern does not define place types as this would be out if scope and limit reusability. Instead, one can state that a certain fix was taken at a position within a recognized place of a given type, e.g., using a subclass of DBpedia's *place* class. The pattern also provides a more general way to add domain and application-specific information by allowing every fix and segment to have additional attributes, e.g., the type of street traveled. Summing up, the pattern provides hooks for other patterns and ontologies thereby acting as a true building block that can be easily extended and combined.

Let us finally consider an example that brings the aforementioned points together. Figure 11.1 shows a part of a human trajectory. It starts at a fix that does not provide any further positioning details, e.g., to mask the home location of a user. The first segment of the trajectory is traveled by a car. Fix1 was taken by the car's navigation system at a specific position that can be represented as a point-feature in Well-Known-Text (WKT) serialization and by using OWL-Time. Segment4 was traveled via a specific subway line. So far, none of the segments had an associated geometry. However, this is possible (conform with GeoSPARQL), e.g., by using a LineString. Finally, the last fix was created by a user's smartphone and the position is not given in terms of geographic coordinates but as so-called geo-social check-in to a Point Of Interest, e.g., a bar, using a social network application such as Foursquare/Swarm. We have since used the Semantic Trajectory pattern to model oceanographic cruises, wildlife tracking, and so forth to substantiate the reusability and extensibility claims. In cases, of cruises, for instance, POI are stops at ports and the moving objects are restricted to research vessels.

11.5. Summary

In this chapter we briefly motivated the need for an ontology design pattern-driven approach to ontology engineering by pointing out that Semantic Web technologies have been often misunderstood as tools to reach or enforce a common agreement, i.e., view on the world, while their true nature is in making meaning and thus differences in the interpretation of terms in different communities explicit. We revisited the *perceived* value proposition of ontologies for domain experts to understand their motivation for joining VoCamps and for using Semantic Web technologies. We pointed out that not all of these hopes can be addressed by the current state-of-the-art or because they contradict with essential design choices made at early stages of the Semantic Web. Next, we discussed how VoCamps are structured and the methods they use to develop patterns. Finally, we have presented a pattern developed at a VoCamp to highlight selected design decisions by example.

Bibliography

[1] C. Bizer, T. Heath, and T. Berners-Lee. Linked data-the story so far. *Semantic Services, Interoperability and Web Applications: Emerging Concepts*, pages 205–227, 2009.

[2] E. Blomqvist, P. Hitzler, K. Janowicz, A. Krisnadhi, T. Narock, and M. Solanki. Considerations regarding ontology design patterns. *Semantic Web*, 7(1):1–7, 2015.

[3] M. Compton, P. Barnaghi, L. Bermudez, R. GarcÃŋa-Castro, O. Corcho, S. Cox, J. Graybeal, M. Hauswirth, C. Henson, A. Herzog, V. Huang, K. Janowicz, W. D. Kelsey, D. L. Phuoc, L. Lefort, M. Leggieri, H. Neuhaus, A. Nikolov, K. Page, A. Passant, A. Sheth, and K. Taylor. The SSN ontology of the W3C semantic sensor network incubator group. *Web Semantics: Science, Services and Agents on the World Wide Web*, 17:25 – 32, 2012.

[4] M. Fernández-López, A. Gómez-Pérez, and N. Juristo. Methontology: from ontological art towards ontological engineering, 1997.

[5] A. Gangemi. A comparison of knowledge extraction tools for the semantic web. In *The Semantic Web: Semantics and Big Data*, pages 351–366. Springer, 2013.

[6] A. Gangemi and V. Presutti. Ontology design patterns. In *Handbook on ontologies*, pages 221–243. Springer, 2009.

[7] M. Grüninger and M. S. Fox. The role of competency questions in enterprise engineering. In *Benchmarking – Theory and Practice*, pages 22–31. Springer, 1995.

[8] Y. Hu, K. Janowicz, D. Carral, S. Scheider, W. Kuhn, G. Berg-Cross, P. Hitzler, M. Dean, and D. Kolas. A geo-ontology design pattern for semantic trajectories. In *Spatial Information Theory*, pages 438–456. Springer, 2013.

[9] P. Jain, P. Hitzler, P. Z. Yeh, K. Verma, and A. P. Sheth. Linked data is merely more data. In *AAAI Spring Symposium: linked data meets artificial intelligence*, volume 11, 2010.

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

[11] K. Janowicz and P. Hitzler. Key ingredients for your next semantics elevator talk. In *Advances in Conceptual Modeling*, pages 213–220. Springer Berlin Heidelberg, 2012.

[12] K. Janowicz, P. Maué, M. Wilkes, S. Schade, F. Scherer, M. Braun, S. Dupke, and W. Kuhn. Similarity as a quality indicator in ontology engineering. In *Proceedings of the 2008 Conference on Formal Ontology in Information Systems: Proceedings of the Fifth International Conference (FOIS 2008)*, pages 92–105, Amsterdam, The Netherlands, The Netherlands, 2008. IOS Press.

[13] 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 modular oceanography ontology. In M. Arenas, O. Corcho, E. Simperl, M. Strohmaier, M. d'Aquin, K. Srinivas, P. 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*, pages 301–309. Springer International Publishing, 2015.

[14] D. B. Lenat, M. Prakash, and M. Shepherd. Cyc: Using common sense knowledge to overcome brittleness and knowledge acquisition bottlenecks. *AI magazine*, 6(4):65, 1985.

[15] H. S. Pinto, S. Staab, and C. Tempich. DILIGENT: Towards a fine-grained methodology for distributed, loosely-controlled and evolving engineering of ontologies. In *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI 2004)*, volume 110, page 393, 2004.

[16] V. Presutti, E. Daga, A. Gangemi, and E. Blomqvist. eXtreme design with content ontology design patterns. In *Proc. Workshop on Ontology Patterns, Washington, DC, USA*. Citeseer, 2009.

[17] S. Scheider and W. Kuhn. How to talk to each other via computers: Semantic interoperability as conceptual imitation. In *Applications of Conceptual Spaces*, pages 97–122. Springer, 2015.

[18] S. Scheider and M. Tomko. Knowing whether spatio-temporal analysis procedures are applicable to datasets. In *9th International Conference on Formal Ontology in Information Systems (FOIS 2016)*. IOS Press, 2016.

[19] P. Spyns, Y. Tang, and R. Meersman. An ontology engineering methodology for dogma. *Applied Ontology*, 3(1-2):13–39, 2008.

[20] R. Studer, V. R. Benjamins, and D. Fensel. Knowledge engineering: principles and methods. *Data & knowledge engineering*, 25(1):161–197, 1998.

[21] M. C. Suárez-Figueroa, A. Gomez-Perez, and M. Fernandez-Lopez. The NeOn methodology for ontology engineering. In *Ontology engineering in a networked world*, pages 9–34. Springer, 2012.

[22] M. Uschold, M. Gruninger, et al. Ontologies: Principles, methods and applications. *Knowledge engineering review*, 11(2):93–136, 1996.