Using SPARQL to describe GIS methods in terms of the questions they answer

Simon Scheider
Universiteit Utrecht/
Dep. of Hum. Geogr. Pl.
Heidelberglaan 2
3584 CS Utrecht,
The Netherlands
simonscheider@web.de

Rob Lemmens
University of Twente/
ITC
Hengelosestraat 99
7514 AE Enschede
The Netherlands
r.l.g.lemmens@utwente.nl

Abstract

GIS methods consist of computational and analytic tools applied to data in order to answer a specific question. For example, in which way is the social wellbeing of a city different from its surroundings? Or, in how far is the surrounding population affected by road construction? The first question may be answered e.g. by constructing a choropleth map, the second one by an area interpolation. In order to effectively search and reuse such GIS methods over the Web, it is necessary to describe them in terms of the questions they answer, not only in terms of particular software or data types. This requires a way to match requests to methods, where both are described in terms of queries about a geographic subject matter, as envisioned in previous work on Datalog based geoservice chaining. However, to truly cover GIS methods, we argue that a much more expressive interrogative Web language is needed, which allows querying over relations and classes and includes completion statements. In this paper, we explain why and discuss in how far SPARQL query containment might be a suitable approach to this end.

Keywords: GIS methods, Question based description, Query containment, SPARQL, linked data.

1 Requesting GIS methods based on the questions they answer

Recently, we saw several initiatives to publish workflows as linked data (Alper et al, 2014; Daga et al, 2014; Belhajjame et al, 2015; Hofer et al, 2016; Scheider and Ballatore 2017). Linked workflows make GIS methods accessible on the Web as a whole as well as on the level of particular tools or data involved (Scheider and Ballatore 2017). This should make it easier for GIS analysts to search, find and exchange their methods, just as they currently exchange their data and code[1] (Rey, 2009; Müller et al, 2013; Bernard et al, 2014). However, this is only true to the extent that we are able to describe these

Table 1: Example methods with corresponding user request and informal question answered by the method.

Example method	Method request	Question answered by method
1 Choropleth classification	I need a method to determine the attribute classes of classification scheme *s* in which the regions from layer *l* lie.	Given a region layer *l* and a classification scheme *s*, in which attribute classes of *s* lie *l*'s regions?
2 Spatial relation check	I need a method to determine whether region r_1 is *contained* in region r_2.	Given two regions r_1 and r_2 and a relation *contained*, is r_1 *contained* in r_2?
3 Spatial relation query	I need a method to determine which geometries *are contained* in region r_2.	Given a region r_2 and a relation *contained*, which geometries are *contained* in it?
4 Area interpolation (MAUP)	I need a method to estimate the aggregation values of region layer *ltarget* using the entities aggregated into region layer *l*.	Given layer *l* with regional aggregations and region layer *ltarget*, which aggregation values do the regions of *ltarget* have using the aggregation of entities in layer *l*?

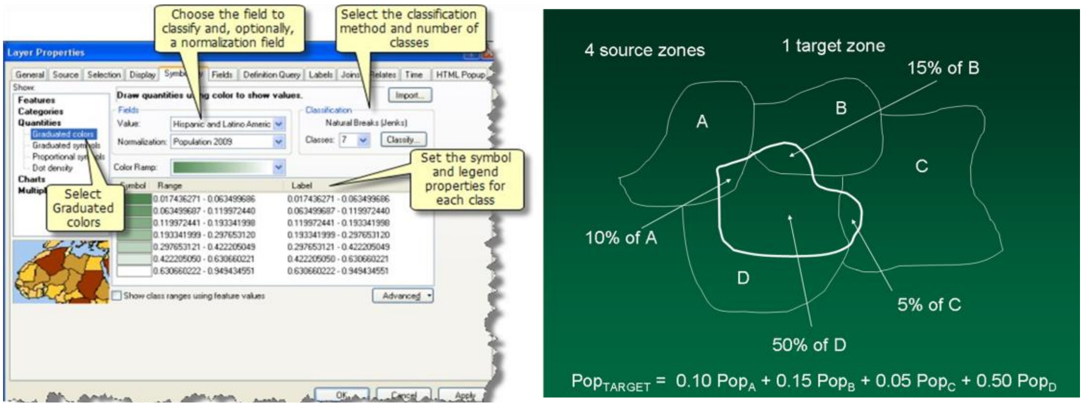
[1] Note, however, there are essential differences between methods and code. Methods, e.g., are easily adapted to new data, while code is not (Hinsen, 2014).

methods in an adequate manner, so that analysts can find what they need.

While linked workflows treat GIS methods in terms of chains of inputs and outputs, they do not easily capture underlying intentions. Current Geoweb service standards such as OGC's Web Processing Services (WPS) mainly rely on textual metadata for this purpose (OGC, 2015). Analytic intentions can be described in natural language, but this approach seems not precise enough to facilitate automatic discovery and matching of methods. Researchers in the Semantic Web and GIScience have therefore been proposing formal service requests[2], focusing on a method's input and outputs, its preconditions and postconditions (Visser et al, 2002; Lemmens et al, 2006; Ludäscher et al, 2006; Lutz, 2007; Fitzner et al, 2011; Brauner, 2015). However, to effectively capture intentions underlying GIS methods, it is necessary to go beyond types of inputs and outputs and types of tools (i.e., a method's type signature) (Hofer et al, 2016). First, different methods can have the same signature and thus cannot be distinguished based on their signature. Second, it is necessary to capture functional dependencies between inputs and outputs which are not easily expressed with data types (Fitzner et al, 2011). And third, users of a method are primarily interested in whether it is capable of answering their question (Kuhn and Ballatore, 2015; Gao and Goodchild, 2013), which is not captured in a method's signature. We illustrate these claims with four examples.

A choropleth map classification method (method 1 in Table 1), as available in ArcMap[3], allows analysts to determine and visually compare the attribute class into which each region of a given spatial layer falls (see Fig. 1a). However, it is usually not interesting for these analysts to know that the method is part of the "layer properties" tab of ArcGIS, because this is just a technical detail of the tool[4]. Furthermore, it is not sufficient to know that the method takes a region layer as input and generates a map, because many mapping techniques do this. As another example, consider the problem of finding out whether some region is contained in another one (method 2 in Table 1), or which regions spatially intersect or overlap with a given one (method 3 in Table 1). The former question can be answered, e.g., by the *SDO_Relate* operator in Oracle Spatial (Oracle, 2017), and the latter one by ArcGIS's "Select by Location" tool[5]. Note however, that different topological operators that might be used for this purpose have exactly identical number and types of input and output (Polygons/Boolean values), and thus cannot be distinguished by their signature (Hofer et al, 2016). Finally, consider a more involved example, which illustrates functional dependencies between inputs and outputs and the difficulty of capturing underlying questions (method 4 in Table 1 and Fig. 1b). Suppose you have a layer of administrative regions with population counts as attribute. Suppose you want to examine how many people are affected by the construction of a road within a certain buffer region. This number can only be estimated, since population data is available only in aggregated form, giving rise to a problem called MAUP[6]. There are different methods to estimate the lacking aggregation, all of them known under the term *areal interpolation* (De Smith et al, 2007, Sect. 4.2.10). The simplest one is to compute a weighted average of all source regions overlapping with a target region, where weights for each source region are determined by its relative coverage of the target

Figure 1: Example methods described in this paper.



(a) Choropleth classification method in ArcGIS.

Layer Properties

Draw quantities using color to show values.

Fields
Value: Hispanic and Latino Americ
Normalization: Population 2009

Classification
Natural Breaks (Jenks)
Classes: 7

Symbol	Range	Label
	0.017436271 - 0.063499686	0.017436271 - 0.063499686
	0.063499687 - 0.119972440	0.063499687 - 0.119972440
	0.119972441 - 0.193341998	0.119972441 - 0.193341998
	0.193341999 - 0.297653120	0.193341999 - 0.297653120
	0.297653121 - 0.422205049	0.297653121 - 0.422205049
	0.422205050 - 0.630660221	0.422205050 - 0.630660221
	0.630660222 - 0.949434551	0.630660222 - 0.949434551

□ Show class ranges using feature values

Source: Environmental Systems Research Institute (ESRI).

(b) Simple areal interpolation method (De Smith et al, 2007).

$$Pop_{TARGET} = 0.10\ Pop_A + 0.15\ Pop_B + 0.05\ Pop_C + 0.50\ Pop_D$$

Source: University of California Santa Barbara (UCSB), GIS course 176B.

[2] Such as WSMO, https://www.w3.org/Submission/WSMO/

[3] http://desktop.arcgis.com/en/arcmap/10.3/map/working-with-layers/a-quick-tour-of-displaying-layers.htm

[4] Which, however, causes headaches when searching in vain through ArcGIS's toolbox.

[5] http://desktop.arcgis.com/en/arcmap/10.3/map/working-with-layers/using-select-by-location.htm

[6] The *modifiable area unit problem* (Jones, 2014, pp.211f).

(assuming homogeneous distribution of the population, see Fig. 1b). Note that all these methods have the basic functional constraint that the source regions together need to fully cover the target region, otherwise the estimation is infeasible (De Smith et al, 2007, Sect. 4.2.10). Furthermore, in their request, analysts ask for an unknown regional aggregation of (hidden) entities, given a different regional aggregation of these same entities (Table 1). Note that this question is not reflected by any of the involved types of tools or geodata. While other authors have described method intentions in terms of underlying spatial concepts (Kuhn and Ballatore, 2015; Kuhn, 2012), operator interrelations (Brauner, 2015), or by handling questions in terms of informal text (Gao and Goodchild, 2013), we focus in this paper on formalizing the questions themselves. In the following, we first give an in-depth analysis of our example questions in the light of existing formal approaches to motivate our linked data based proposal.

2 Formalizing the question answered by a method

Lutz (2007) and Fitzner et al (2011) have emphasized the need for capturing functional dependencies between a method's inputs and outputs and proposed to represent GIS methods in terms of queries using Horn rules/Datalog, of the form[7]:

Rule (body → head):

$$\forall x,\ldots,z.\ P_1(x,\ldots,y) \wedge \ldots \wedge P_n(w,\ldots,z) \rightarrow P_h(x,\ldots,z) \qquad (1)$$

Conjunctive query (no head):

$$P_1(?x,\ldots,?y) \wedge \ldots \wedge P_n(?w,\ldots,?z) \qquad (2)$$

When both requests (R) and methods (M) are represented as Datalog queries, then it becomes possible to match them in an efficient way based on query containment, i.e., testing whether one query contains the other (Lemmens, 2006, Sect. 6.4) ($M \subseteq R$):

Definition. *A query Q_1 is contained in a query Q_2, written $Q_1 \subseteq Q_2$, if the set of facts obtained from Q_1 is a subset of facts obtained from Q_2.*

For example, if we request for an overlay operation with two spatial regions as inputs (?x,?y) and one region as output (?z), then this strategy would return the method "intersection", since intersection is subsumed by overlay (in a GIS ontology: *Intersect(?x, ?y, ?z) → Overlay(?x, ?y, ?z)*), i.e., all results returned by an intersection query are also overlay results (see also Lemmens, 2006, pp. 173):

Request query:

$$\mathit{Region}(?x) \wedge \mathit{Region}(?y) \wedge \mathit{Region}(?z) \wedge \mathit{Overlay}(?x, ?y, ?z) \qquad (3)$$

Method query:

$$\mathit{Region}(?x) \wedge \mathit{Region}(?y) \wedge \mathit{Region}(?z) \wedge \mathit{Intersect}(?x, ?y, ?z) \qquad (4)$$

Note that functional dependencies between inputs and outputs, such as ?z being the intersection of ?x and ?y, are needed to express method constraints and can be captured by n-ary predicates. The advantage of handling questions with query containment is that we do not have to know the answer (the query result) in order to know whether a method is useful for answering them. Datalog was chosen because it can express rules and because query containment can be computed efficiently. As can be seen above, however, Datalog has important syntactic restrictions:

1. Variables range only over instances, and never over predicates (= classes or relations)
2. Any variable in the head of a rule must also appear in the body (no existential quantification in the head, i.e., no expressions of the form $\forall \rightarrow \exists$)
3. Queries allow only a very restricted form of (”stratified”) negation (¬)

While these restrictions make Datalog reasoning as well as query containment efficiently computable, they also mean our example methods cannot be adequately described. Suppose we want to express the method of example 1 (choropleth classification):

$$\mathit{hasAttrValue}(?l,?a) \wedge \mathit{ClassofScheme}(?\mathit{Class},?s) \wedge ?\mathit{Class}(?a) \qquad (5)$$

Paraphrased, this means: We are looking for the class of a given classification scheme that applies to the attribute value of a given layer. However, this requires a variable *?Class* that ranges over predicates, not instances (contradicting restriction 1). Furthermore, suppose we want to express *examples 2 and 3* (spatial relation check and query) as a query:

$$\mathit{Region}(?r_1) \wedge \mathit{Region}(?r_2) \wedge ?\mathit{XRelate}(?r_1, ?r_2) \qquad (6)$$

Paraphrased, this means: We test a given relation between two given regions, or search for the regions that are related in a given way to a given region. Note that the spatial relation is a free variable, not a constant, resulting in the same dilemma (contradicting restriction 1). Furthermore, consider the *4th example* (areal interpolation):

$$\mathit{Layer}(?\mathit{ltgt}) \wedge \mathit{Layer}(?l) \wedge \exists ?\mathit{lgrd}.\ \mathit{AggrOf}(?l, ?\mathit{lgrd}) \wedge \mathit{AggrOf}(?\mathit{ltgt}, ?\mathit{lgrd}) \qquad (7)$$

Paraphrased, this means: We are looking for a target layer (*ltgt*) aggregated from a (implicit) ground layer (*lgrd*) from which our input layer *l* was aggregated. This can be expressed with a conjunctive query. However, we also need to account for the following precondition:

$$\forall r.\ \mathit{hasRegion}(?\mathit{ltgt}, r) \rightarrow \exists r'.\ \mathit{MergedRegionOf}(r', ?l) \wedge \mathit{Contains}(r', r) \qquad (8)$$

Paraphrased, this means: all regions of a given target layer must be contained in the merger of the regions of a given input layer.

[7] We express formulas in a logic form. Variables are denoted by x,...,z, and P_i denote predicates. Variables can be bound by quantifiers (∀, ∃), and formulas are combined by logical connectives (∧,→). In what follows, we express free (unbound) variables with a preceding question mark, *?x*, following the SPARQL convention.

Listing 1: Choropleth classification

```
SELECT ?class WHERE {
?l ada:hasElement ?e. ?e ada:hasRegion ?r; ada:hasMeasure ?a.
?class ada:classOfScheme ?s. ?a a ?class . }
```

Listing 2: Relation check

```
ASK {
?r1 a gis:Region . ?xrelate rdfs:subPropertyOf gis:spatialRelation.
?r2 a gis:Region. ?r1 ?xrelate ?r2 . FILTER(?r1 != ?r2)}
```

Listing 3: Relation query

```
SELECT ?r2 WHERE {
?r1 a gis:Region . ?xrelate rdfs:subPropertyOf gis:spatialRelation.
?r2 a gis:Region. ?r1 ?xrelate ?r2 . FILTER(?r1 != ?r2)}
```

Listing 4: Areal interpolation

```
SELECT ?vlsttgt WHERE {
?l a gis:Layer. ?ltgt a gis:Layer.
?vlst ada:ofDataset ?l; ada:ofAttr ?hasMsr; a ada:ValueList.
?vlst gis:isAggrOf ?lgrd.
?vlsttgt ada:ofDataset ?ltgt; ada:ofAttr ?hasMsr; a ada:ValueList.
?vlsttgt     gis:isAggrOf     ?lgrd.
FILTER(?l != ?ltgt)
FILTER NOT EXISTS { ?ltgt ada:hasElement ?i. ?i ada:hasRegion ?rtgt.
FILTER NOT EXISTS {
?rm geo:sfContains ?rtgt. ?rm gis:mergedRegionOf ?l.}}
}
```

Note that this condition can neither be expressed by a data type nor any functional description, but only by a *completion*[8] statement which requires existential quantification in the rule head (*∀→∃*, "for all ... there exists ..."). However, this contradicts restrictions 2 and 3.

3 Describing GIS methods in terms of SPARQL queries

SPARQL 1.1 is the query language of the Semantic Web[9]. In contrast to Datalog, it is based on a logic with a very expressive formal semantics, the Resource Description Framework (RDF), which is also the basis for linked data. Linked data cconsists of *subject - predicate – object* statements (triples), where the predicate is an arrow linking subject to object (see Fig. 2). In effect, RDF is a higher-order language (Hitzler et al, 2009). While Semantic Web reasoning languages such as OWL2 profiles[10] and RDFS are first-order to stay within decidable bounds, SPARQL can afford to be higher order, since it is not a language for reasoning. There are three features of SPARQL/RDF that make it a candidate for solving our method description problem:

1. As a higher-order language, it allows quantification over relations and classes. Relations are the "arrows" in linked data triples, and classes are linked to instances via the relation *rdf:type*, abbreviated by *a*.
2. It allows distinguishing *bound* and *unbound* variables to tell goals (what you want to know) from other unknowns in a method. Bound variables are inside a SELECT clause.
3. It allows expressing completion (*∀→∃*) in terms of two nested FILTER NOT EXISTS statements, which corresponds to a logical statement of the form ¬*∃*¬*∃*, a twofoldly negated existentially quantified query ("it is not the case that not").

Using this apparatus, we can reformulate our 4 method queries as follows[11], building on the vocabularies *AnalysisData*[12], *GISConcepts*[13] and *GeoSparql*[14] to represent layers (datasets) in terms of their data elements. Data elements link a single region to some measure (an attribute value), see also Fig. 2 (Scheider and Tomko, 2016). *Outputs* of the method (the question's goals) are encoded as bound variables in the SELECT clause or in terms of an ASK clause. The latter denotes a question answered by yes or no. The *input parameters* of the method are a subset of the free variables. For

[8] With a completion statement, we require all things of a certain kind to stand in a relation to some other entities.

[9] https://www.w3.org/TR/sparql11-overview/

[10] Decidable subsets of the Web Ontology Language, https://www.w3.org/TR/owl2-profiles/

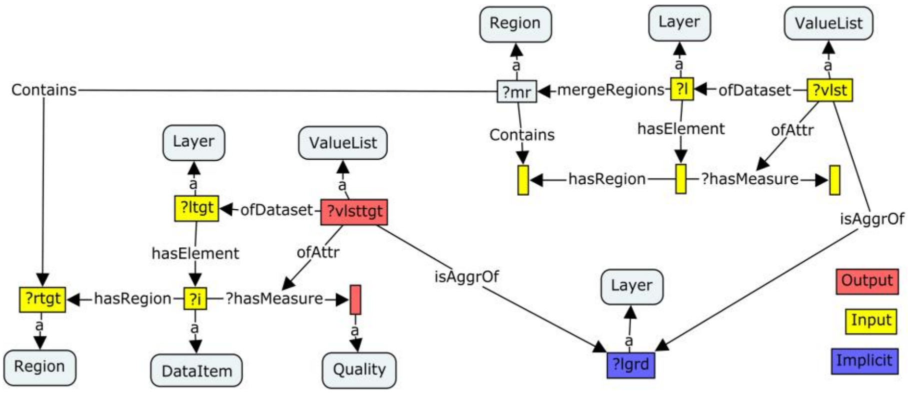
[11] We successfully tested these queries on data examples using https://github.com/simonscheider/meaningfulSTAnalytics

[12] *ada:* http://geographicknowledge.de/vocab/AnalysisData.rdf

[13] *gis:* http://geographicknowledge.de/vocab/GISConcepts.rdf

[14] *geo:* http://www.opengis.net/ont/geosparql

Figure 2: SPARQL query pattern for describing areal interpolation methods. Arrows denote relations, rectangles instances, and rounded rectangles are classes. Colours denote method inputs, outputs and implicit information. See Listing 4 and explanation in text.



query matching, these can be marked with a particular naming scheme and substituted with the parameters of a request. In the choropleth classification case (Listing 1), we query over the classes of a particular class scheme *?s*, given as parameter to the method, together with a layer *?l* and a region *?r*. In the relation check (Listing 2) and the relation query (Listing 3), we query with a spatial relation given as parameter together with either one or two regions. This is only possible because SPARQL deals easily with variables over predicates and classes. The output is either a set of regions or a boolean value, where the latter is captured in terms of the ASK clause. Note that we require input regions to be distinct using a FILTER expression.

The areal interpolation query (Listing 4) is a more complex case (compare Fig. 2). The basic graph pattern (the set of triples in the where clause not in the FILTER statement) includes a description of what outcome is expected: namely a situation where we have two distinct region layers as input (*?l* (=”source layer”) and *?ltgt* (=”target layer”)), both of which have an attribute which is an aggregation of some other (implicit) layer (shown in blue in Fig. 2) that is assumed to exist in the back-ground (*?lgrd* =”ground layer”). GIS attributes are expressed using the pattern of Scheider and Tomko (2016), namely as a *ValueList* of some measure of data elements of the layer (Fig. 2). The layer *?l* comes with an aggregated attribute, while the aggregated attribute *?vlsttgt* of the target layer is the requested output. The nested negated FILTER statement expresses the completion precondition on the input layers. For this purpose, we use a relation *gis:mergeRegions*, which links layers to the merger of the set of their regions[15]. The latter must contain the target regions[16].

4 Open questions and future work

One central question then is in how far deciding whether a SPARQL query is contained in another is a feasible computational task. Unfortunately, due to its expressiveness, this problem is undecidable for *full* SPARQL (Pichler and Skritek, 2014). However, it has been shown that interesting fragments such as conjunctive queries without projection and acyclic queries[17] without OPTIONAL statements can be solved efficiently (Chekol et al, 2013, 2011). Our paper illustrates that basic (acyclic) graph patterns can capture higher order constraints in GIS methods, and that additional FILTER statements can express completion and relational constraints. Future work could therefore concentrate on query containment for these types of patterns. A particularly interesting option is to turn basic graph patterns with FILTER NOT EXISTS statements into RDF data sets that are satisfied by the query (using blank nodes for variables and adding triples up to completion), and then to test containment simply by querying over this data. Another problem that needs to be solved to this end is the *parameterization* of the method query with inputs from the request query.

5 Conclusion

We showed that even for rather simple GIS methods as the ones considered in this paper, capturing analytic intentions and preconditions in terms of questions requires not only functional constraints over inputs and outputs, but also higher order variables as well as complex completion statements. While the idea of query containment and matching as suggested in the past allows requesting methods in terms of the questions they answer, Datalog and Horn rule bases as well as OWL profiles are not sufficiently expressive for this purpose. SPARQL seems to cope well in terms of basic (RDF) graph patterns and (nested) FILTER statements. Future work should therefore address query containment for respective SPARQL patterns.

6 References

Alper P, Belhajjame K, Goble CA, Karagoz P (2014) Labelflow: Exploiting workflow provenance to surface

[15] https://desktop.arcgis.com/en/arcmap/latest/extensions/production-mapping/merging-geometries.htm

[16] In GeoSPARQL, this can also be expressed in terms of a separate FILTER statement (Battle and Kolas, 2012)

[17] Where the graph pattern does not contain a cycle, see Chekol et al (2013).

scientific data provenance. In: International Provenance and Annotation Workshop, Springer, pp 84–96

Battle R, Kolas D (2012) Enabling the geospatial semantic web with parliament and geosparql. Semantic Web 3(4):355–370

Belhajjame K, Zhao J, Garijo D, Gamble M, Hettne K, Palma R, Mina E, Corcho O, Gomez-Perez JM, Bechhofer S, Klyne G, Goble C (2015) Using a suite of ontologies for preserving workflow-centric research objects. Web Semantics 32:16–42

Bernard L, Mäs S, Müller M, Henzen C, Brauner J (2014) Scientific geodata infrastructures: challenges, approaches and directions. International Journal of Digital Earth 7(7):613–633

Brauner J (2015) Formalizations for geooperators - geoprocessing in spatial data infrastructures. PhD thesis, TU Dresden

Chekol MW, Euzenat J, Genevès P, Layaīda N (2011) PSPARQL query containment. In: Foster N, Kementsietsidis A (eds) Database Programming Languages - DBPL 201, 13th International Symposium, Seattle, Washington, USA, August 29, 2011. Proceedings

Chekol MW, Euzenat J, Genevès P, Layaīda N (2013) Evaluating and benchmarking sparql query containment solvers. In: International Semantic Web Conference, Springer, pp 408–423

Daga E, dAquin M, Gangemi A, Motta E (2014) Describing semantic web applications through relations between data nodes. Tech. rep., Citeseer

De Smith MJ, Goodchild MF, Longley P (2007) Geospatial analysis: a comprehensive guide to principles, techniques and software tools. Troubador Publishing Ltd

Fitzner D, Hoffmann J, Klien E (2011) Functional description of geoprocessing services as conjunctive datalog queries. Geoinformatica 15(1):191–221

Gao S, Goodchild MF (2013) Asking spatial questions to identify GIS functionality. In: Computing for Geospatial Research and Application (COM. Geo), 2013 Fourth International Conference on, IEEE, pp 106–110

Hinsen K (2014) Computational science: shifting the focus from tools to models. F1000Research 3(101)

Hitzler P, Krötzsch M, Rudolph S (2009) Foundations of Semantic Web Technologies. Chapman & Hall/CRC

Hofer B, Mäs S, Brauner J, Bernard L (2016) Towards a knowledge base to support geoprocessing workflow development. International Journal of Geographical Information Science pp 1–23

Jones CB (2014) Geographical information systems and computer cartography. Routledge

Kuhn W (2012) Core concepts of spatial information for transdisciplinary research. International Journal of Geographical Information Science 26(12):2267–2276

Kuhn W, Ballatore A (2015) Designing a language for spatial computing. In: AGILE 2015, Springer, pp 309–326

Lemmens R, Wytzisk A, By R, Granell C, Gould M, van Oosterom P (2006) Integrating semantic and syntactic descriptions to chain geographic services. IEEE Internet Computing 10(5):42–52

Lemmens RL (2006) Semantic interoperability of distributed geo-services. PhD thesis, TU Delft, Delft University of Technology

Ludascher B, Lin K, Bowers S, Jaeger-Frank E, Brodaric B, Baru C (2006) Managing scientific data: From data integration to scientific workflows. Geological Society of America Special Papers 397:109–129

Lutz M (2007) Ontology-based descriptions for semantic discovery and composition of geoprocessing services. GeoInformatica 11(1):1–36

Müller M, Bernard L, Kadner D (2013) Moving code–sharing geoprocessing logic on the web. ISPRS journal of photogrammetry and remote sensing 83:193–203

OGC (2015) Ogc wps 2.0 interface standard. ogc document 14-065. Tech. rep., Open Geospatial Consortium, Wayland, MA

Oracle (2017) Oracle Big Data Spatial and Graph. [Online; accessed Jan 2017] http://www.oracle.com/database/big-data-spatial-and-graph

Pichler R, Skritek S (2014) Containment and equivalence of well-designed sparql. In: Proceedings of the 33rd ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, ACM, pp 39–50

Rey SJ (2009) Show me the code: spatial analysis and open source. Journal of Geographical Systems 11(2):191–207

Scheider S, Ballatore A (2017) Semantic typing of linked geoprocessing workflows. International Journal of Digital Earth: 1-29

Scheider S, Tomko M (2016) Knowing whether spatio-temporal analysis procedures are applicable to datasets. In: Formal Ontology in Information Systems - Proceedings of the 9th International Conference, FOIS 2016, Annecy, France, July 6-9, 2016, pp 67–80

Visser U, Stuckenschmidt H, Schuster G, Vogele T (2002) Ontologies for geographic information processing. Computers & Geosciences 28:103–117